

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problems Mailbox.**



PATENT ABSTRACTS OF JAPAN

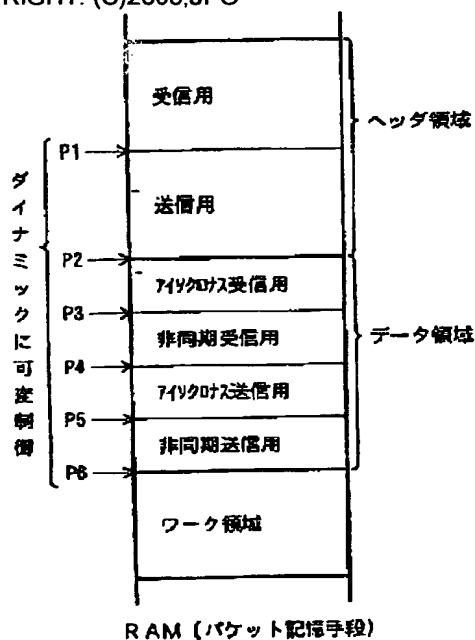
(11) Publication number: **2000134233 A**(43) Date of publication of application: **12 . 05 . 00**(51) Int. Cl. **H04L 12/40**(21) Application number: **10321541**(22) Date of filing: **27 . 10 . 98**(71) Applicant: **SEIKO EPSON CORP**(72) Inventor: **OGAWA TAKAHISA
ISHIDA TAKUYA
KANBARA YOSHIYUKI**(54) **DATA TRANSFER CONTROLLER AND ELECTRONIC EQUIPMENT**

COPYRIGHT: (C)2000,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a data transfer controller and electronic equipment capable of reducing the overheads of processings and transferring data at a high speed with the hardware of a small scale.

SOLUTION: In this data transfer controller of an IEEE1394 standard, between a link core and a CPU, a built-in RAM for storing packets capable of random access is provided other than a FIFO. The storage area of the RAM is separated into a header area, a data area and the work area of the CPU and the header area and the data area are separated into areas for reception and for transmission. By using TAG, the header of a reception packet is written in the header area and the data are written in the data area. The data area is separated into the areas for isochronous transfer and for asynchronous transfer. A pointer for variably controlling the size of the respective areas of the RAM is prepared and the size of the respective areas is dynamically and variably controlled even after supplying power. The respective areas are turned to a ring buffer structure. The size of the area for storing the header and data of one packet is fixed.



RAM (パケット記憶手段)

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-134233

(P2000-134233A)

(43) 公開日 平成12年5月12日 (2000.5.12)

(51) Int.Cl.⁷

H 0 4 L 12/40

識別記号

F I

H 0 4 L 11/00

テマコード* (参考)

3 2 0

5 K 0 3 2

審査請求 未請求 請求項の数19 F D (全 29 頁)

(21) 出願番号

特願平10-321541

(22) 出願日

平成10年10月27日 (1998. 10. 27)

(71) 出願人

000002369

セイコーエプソン株式会社

東京都新宿区西新宿 2 丁目 4 番 1 号

(72) 発明者

小川 隆央

長野県諏訪市大和 3 丁目 3 番 5 号 セイコ

ーエプソン株式会社内

(72) 発明者

石田 卓也

長野県諏訪市大和 3 丁目 3 番 5 号 セイコ

ーエプソン株式会社内

(74) 代理人

100090479

弁理士 井上 一 (外 2 名)

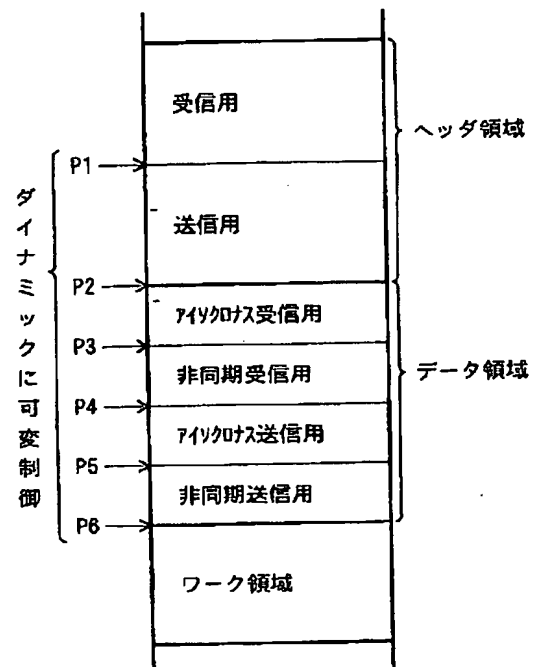
最終頁に続く

(54) 【発明の名称】 データ転送制御装置及び電子機器

(57) 【要約】

【課題】 処理のオーバーヘッドを軽減し、小規模なハードウェアで高速なデータ転送を実現できるデータ転送制御装置及び電子機器を提供すること。

【解決手段】 IEEE 1394 規格のデータ転送制御装置において、リンクコアと CPU との間に、FIFO の他に、ランダムアクセス可能なパケット格納用の内蔵 RAM を設ける。RAM の記憶領域をヘッダ領域、データ領域、CPU のワーク領域に分離し、ヘッダ領域及びデータ領域を受信用と送信用の領域に分離する。TAG を用いて、受信パケットのヘッダをヘッダ領域にデータをデータ領域に書き込む。データ領域をアイソクロナス転送用と非同期転送用の領域に分離する。RAM の各領域のサイズを可変に制御するためのポインタを用意し、各領域のサイズを電源投入後もダイナミックに可変制御する。各領域をリングバッファ構造にする。1 つのパケットのヘッダやデータが格納される領域のサイズを固定にする。



RAM (パケット記憶手段)

【特許請求の範囲】

【請求項 1】 バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、ノード間でのパケット転送のためのサービスを提供するリンク手段と、パケットを格納するためのランダムアクセス可能な記憶手段と、前記リンク手段を介して各ノードから転送されてくるパケットを前記記憶手段に書き込む書き込み手段と、上層により前記記憶手段に書き込まれたパケットを読み出し、前記リンク手段に渡す読み出し手段とを含むことを特徴とするデータ転送制御装置。

【請求項 2】 請求項 1 において、前記記憶手段が、パケットの制御情報が格納される制御情報領域と、パケットのデータが格納されるデータ領域とに分離されていることを特徴とするデータ転送制御装置。

【請求項 3】 請求項 1 において、前記記憶手段が、パケットが格納される領域と中央処理ユニットのワーク領域とに分離されていることを特徴とするデータ転送制御装置。

【請求項 4】 請求項 2 又は 3 のいずれかにおいて、前記記憶手段の前記制御情報領域が、受信用制御情報領域と送信用制御情報領域とに分離されていることを特徴とするデータ転送制御装置。

【請求項 5】 請求項 2 乃至 4 のいずれかにおいて、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むパケット分離手段を含むことを特徴とするデータ転送制御装置。

【請求項 6】 請求項 5 において、前記リンク手段が、パケットの少なくとも制御情報とデータを区別するためのタグ情報を生成すると共に、生成された該タグ情報をパケットに関連づけ、前記パケット分離手段が、パケットに関連づけられた前記タグ情報に基づいて、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むことを特徴とするデータ転送制御装置。

【請求項 7】 請求項 2 乃至 6 のいずれかにおいて、前記記憶手段の前記データ領域が、受信用データ領域と送信用データ領域とに分離されていることを特徴とするデータ転送制御装置。

【請求項 8】 請求項 2 乃至 7 のいずれかにおいて、前記記憶手段の前記データ領域が、アイソクロナス転送用データ領域と非同期転送用データ領域とに分離されていることを特徴とするデータ転送制御装置。

【請求項 9】 請求項 2 乃至 8 のいずれかにおいて、

前記記憶手段の前記データ領域が非同期転送用データ領域を含む、

前記非同期転送用データ領域が、第 1、第 2 の非同期転送用データ領域を含む複数の領域に分離されていることを特徴とするデータ転送制御装置。

【請求項 10】 請求項 2 乃至 9 のいずれかにおいて、前記記憶手段の前記データ領域がアイソクロナス転送用データ領域と第 1 の非同期転送用データ領域とに分離され、

10 前記アイソクロナス転送用データ領域が第 2 の非同期転送用データ領域として使用されることを特徴とするデータ転送制御装置。

【請求項 11】 請求項 2 乃至 10 のいずれかにおいて、

前記記憶手段の前記データ領域が複数の領域に分離され、パケットの制御情報に基づき、該分離された複数の領域のいずれかにパケットのデータを書き込む手段を含むことを特徴とするデータ転送制御装置。

20 【請求項 12】 請求項 1 乃至 11 のいずれかにおいて、前記記憶手段が複数の領域に分離される場合において、各領域のサイズを可変に制御する手段を含むことを特徴とするデータ転送制御装置。

【請求項 13】 請求項 12 において、各領域のサイズが、電源投入後もダイナミックに可変制御されることを特徴とするデータ転送制御装置。

30 【請求項 14】 請求項 1 乃至 13 のいずれかにおいて、前記記憶手段が複数の領域に分離される場合において、分離された領域の一方の境界から他方の境界に向かってパケットの制御情報及びデータの少なくとも一方が格納され、他方の境界に達した場合には一方の境界に戻ってパケットの制御情報及びデータの少なくとも一方が格納されることを特徴とするデータ転送制御装置。

【請求項 15】 請求項 14 において、1つのパケットの制御情報及びデータの少なくとも一方が格納される領域のサイズが固定となることを特徴とするデータ転送制御装置。

40 【請求項 16】 請求項 1 乃至 15 のいずれかにおいて、前記リンク手段と前記記憶手段との間に設けられる受信用 F I F O と送信用 F I F O とを含むことを特徴とするデータ転送制御装置。

【請求項 17】 請求項 1 乃至 16 のいずれかにおいて、IEEE 1394 の規格に準拠したデータ転送を行うことを特徴とするデータ転送制御装置。

50 【請求項 18】 請求項 1 乃至 17 のいずれかのデータ転送制御装置と、

前記データ転送制御装置及びバスを介して他のノードから受信したデータに所与の処理を施す装置と、処理が施されたデータを出力又は記憶するための装置とを含むことを特徴とする電子機器。

【請求項 19】 請求項 1 乃至 17 のいずれかのデータ転送制御装置と、

前記データ転送制御装置及びバスを介して他のノードに送信するデータに所与の処理を施す装置と、

処理が施されるデータを取り込むための装置とを含むことを特徴とする電子機器。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、データ転送制御装置及びこれを含む電子機器に関する。

【0002】

【背景技術及び発明が解決しようとする課題】 近年、IEEE1394 と呼ばれるインターフェース規格が脚光を浴びている。この IEEE1394 は、次世代のマルチメディアにも対応可能な高速シリアルバスインターフェースを規格化したものである。この IEEE1394 によれば、動画像などのリアルタイム性が要求されるデータも扱うことができる。また、IEEE1394 のバスには、プリンタ、スキャナ、CD-R ドライブ、ハードディスクドライブなどのコンピュータの周辺機器のみならず、ビデオカメラ、VTR、TV などの家庭用電化製品も接続できる。このため、電子機器のデジタル化を飛躍的に促進できるものとして期待されている。

【0003】 このような IEEE1394 の概要については、例えば「IEEE1394 ハイ・パフォーマンス・シリアルバスの概要」(Interface Apr. 1996 の 1~10 頁)、「PC 周辺機器用バス規格群総覧」(Interface Jan. 1997 の 106 頁~116 頁)、「IEEE1394-1995 (FireWire) のリアルタイム転送モードとマルチメディア対応プロトコル」(Interface Jan. 1997 の 136~146 頁)に開示されている。また、IEEE1394 に準拠したデータ転送制御装置としては、テキサス・インスツルメンツ社製の TSB12LV31 などが知られている。

【0004】 しかしながら、この IEEE1394 に準拠したデータ転送制御装置には次のような課題があることが判明した。

【0005】 即ち、現在の IEEE1394 規格によれば最大で 400Mbps の転送速度が実現可能となっている。しかし、現実には、処理のオーバーヘッドの存在に起因して、システム全体の実転送速度はこれよりもかなり低くなっている。つまり、CPU 上で動作するファームウェアやアプリケーションソフトが、送信データを準備したり、受信データを取り込んだりするなどの処理に多くの時間を要してしまい、バス上での転送速度が速

くても、結局、高速なデータ転送を実現できない。

【0006】 特に、周辺機器に組み込まれる CPU は、パーソナルコンピュータなどのホストシステムに組み込まれる CPU に比べて処理能力が低い。このため、ファームウェアやアプリケーションソフトの処理のオーバーヘッドの問題は、非常に深刻なものとなる。従って、このようなオーバーヘッドの問題を効果的に解消できる技術が望まれている。

【0007】 本発明は、以上のような技術的課題に鑑みてなされたものであり、その目的とするところは、ファームウェアやアプリケーションソフトなどの処理のオーバーヘッドを軽減し、小規模なハードウェアで高速なデータ転送を実現できるデータ転送制御装置及びこれが用いられる電子機器を提供することにある。

【0008】

【課題を解決するための手段】 上記課題を解決するために本発明は、バスに接続される複数のノード間でのデータ転送のためのデータ転送制御装置であって、ノード間でのパケット転送のためのサービスを提供するリンク手段と、パケットを格納するためのランダムアクセス可能な記憶手段と、前記リンク手段を介して各ノードから転送されてくるパケットを前記記憶手段に書き込む書き込み手段と、上層により前記記憶手段に書き込まれたパケットを読み出し、前記リンク手段に渡す読み出し手段とを含むことを特徴とする。

【0009】 本発明によれば、各ノードから転送されてくるパケットは、書き込み手段によりランダムアクセス可能な記憶手段に書き込まれる。また、ファームウェアやアプリケーションソフトなどの上層により記憶手段に書き込まれたパケットは、読み出し手段により読み出され、リンク手段に渡される。そして、各ノードに転送される。このように本発明では、パケット格納用のランダムアクセス可能な記憶手段を、リンク手段と上層との間に介在させている。このようにすることで、パケットの受信順序や送信順序に依存せずに、記憶手段の所望の記憶領域にパケットを格納することが可能になる。これにより、パケットを、記憶手段の複数の領域に分離して格納することも可能になる。

【0010】 また本発明は、前記記憶手段が、パケットの制御情報が格納される制御情報領域と、パケットのデータが格納されるデータ領域とに分離されていることを特徴とする。このようにすることで、ファームウェアやアプリケーションソフトなどの上層の処理負荷を軽減でき、システム全体の実転送速度を向上できる。また、記憶手段からのパケットの読み出し処理や記憶手段への書き込み処理を簡素化できる。

【0011】 また本発明は、前記記憶手段が、パケットが格納される領域と中央処理ユニットのワーク領域とに分離されていることを特徴とする。このようにすることで、中央処理ユニットのローカルメモリの記憶容量を軽

減したり、ローカルメモリの必要性自体を無くすることが可能になる。

【0012】また本発明は、前記記憶手段の前記制御情報領域が、受信用制御情報領域と送信用制御情報領域とに分離されていることを特徴とする。このようにすることで、受信用制御情報領域から制御情報を連続して読み出したり、送信用制御情報領域に制御情報を連続して書き込むことが可能になり、処理の簡素化、処理負担の軽減を図れる。

【0013】また本発明は、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むパケット分離手段を含むことを特徴とする。このようにすれば、パケットの制御情報とデータが自動的に、各々、制御情報領域、データ領域に書き込まれるようになり、ファームウェアなどの上層の処理のオーバーヘッドを軽減できる。

【0014】また本発明は、前記リンク手段が、パケットの少なくとも制御情報とデータを区別するためのタグ情報を生成すると共に、生成された該タグ情報をパケットに関連づけ、前記パケット分離手段が、パケットに関連づけられた前記タグ情報に基づいて、パケットの制御情報を前記制御情報領域に書き込み、パケットのデータを前記データ領域に書き込むことを特徴とする。このようにすることで、簡易なハードウェア構成で、パケットの制御情報を制御情報領域にデータをデータ領域に格納することが可能になる。

【0015】また本発明は、前記記憶手段の前記データ領域が、受信用データ領域と送信用データ領域とに分離されていることを特徴とする。このようにすることで、受信用データ領域から連続して受信データを読み出したり、送信用データ領域に連続して送信データを書き込むことが可能になり、上層の処理のオーバーヘッドを軽減できる。

【0016】また本発明は、前記記憶手段の前記データ領域が、アイソクロナス転送用データ領域と非同期転送用データ領域とに分離されていることを特徴とする。このようにすれば、非同期パケットの後にアイソクロナスパケットが転送されてきた場合にも、アイソクロナスパケットを先に処理することなどが可能になる。これにより、アイソクロナス転送に要求される処理のリアルタイム性を維持できるようになる。

【0017】また本発明は、前記記憶手段の前記データ領域が非同期転送用データ領域を含み、前記非同期転送用データ領域が、第1、第2の非同期転送用データ領域を含む複数の領域に分離されていることを特徴とする。このようにすれば、非同期転送用データ領域の複数の領域の各々に、用途の異なるデータを別々に格納できるようになる。即ち、例えば、第1の非同期転送用データ領域に、コマンドデータやステータスデータなどの制御用データを格納し、第2の非同期転送用データ領域に、ア

プリケーションにより出力、記憶、取り込み等されるデータを格納することが可能になる。従って、例えば、アプリケーションにより出力、記憶、取り込み等されるデータを、例えば第2の非同期転送用データ領域に連続して格納することが可能になり、上層の処理負担を軽減できるようになる。

【0018】また本発明は、前記記憶手段の前記データ領域がアイソクロナス転送用データ領域と第1の非同期転送用データ領域とに分離され、前記アイソクロナス転送用データ領域が第2の非同期転送用データ領域として使用されることを特徴とする。このようにすれば、アプリケーションがアイソクロナス転送を行わない場合等において、アイソクロナス転送用データ領域を第2の非同期転送用データ領域として使用できるようになる。そして、用途の異なるデータを、第1、第2の非同期転送用データ領域に別々に格納できるようになる。この結果、限られたリソースの有効利用を図りながら、上層の処理負担を軽減できる。

【0019】また本発明は、前記記憶手段の前記データ領域が複数の領域に分離され、パケットの制御情報に基づき、該分離された複数の領域のいずれかにパケットのデータを書き込む手段を含むことを特徴とする。このようにすれば、用途の異なるデータを、分離された複数の領域に別々に格納できるようになる。この結果、上層の処理負担の軽減化を図れる。

【0020】また本発明は、前記記憶手段が複数の領域に分離される場合において、各領域のサイズを可変に制御する手段を含むことを特徴とする。このようにすれば、アプリケーションに応じた最適な領域分割を実現でき、限られたリソースの有効利用を図れる。

【0021】また本発明は、各領域のサイズが、電源投入後もダイナミックに可変制御されることを特徴とする。このようにすれば、受信処理と送信処理が混在するような場合にも、限られたリソースを有効利用することが可能になる。

【0022】また本発明は、前記記憶手段が複数の領域に分離される場合において、分離された領域の一方の境界から他方の境界に向かってパケットの制御情報及びデータの少なくとも一方が格納され、他方の境界に達した場合には一方の境界に戻ってパケットの制御情報及びデータの少なくとも一方が格納されることを特徴とする。このようにすれば、ランダムアクセス可能な記憶手段にFIFOの機能を持たせることが可能になる。また他方の境界に達した場合に一方の境界からパケットの制御情報やデータを格納できるため、限られたリソースを有効利用できる。

【0023】また本発明は、1つのパケットの制御情報及びデータの少なくとも一方が格納される領域のサイズが固定となることを特徴とする。このようにすることで、ファームウェアなどによるパケットの取り扱いが簡

素化され、ファームウェアなどの処理負担を軽減できる。

【0024】なお、本発明では、前記リンク手段と前記記憶手段との間に設けられる受信用FIFOと送信用FIFOとを含むようにすることができる。

【0025】また、本発明では、IEEE1394の規格に準拠したデータ転送を行うことが望ましい。

【0026】また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードから受信したデータに所与の処理を施す装置と、処理が施されたデータを出力又は記憶するための装置とを含むことを特徴とする。また本発明に係る電子機器は、上記のいずれかのデータ転送制御装置と、前記データ転送制御装置及びバスを介して他のノードに送信するデータに所与の処理を施す装置と、処理が施されるデータを取り込むための装置とを含むことを特徴とする。

【0027】本発明によれば、他のノードから転送されたデータを電子機器において出力したり記憶したりする処理、電子機器において取り込んだデータを他のノードに転送したりする処理を高速化することが可能になる。また、本発明によれば、データ転送制御装置を小規模化できると共に、データ転送を制御するファームウェアなどの処理負担を軽減できるため、電子機器の低コスト化、小規模化などを図ることも可能になる。

【0028】

【発明の実施の形態】以下、本発明の好適な実施形態について図面を用いて詳細に説明する。

【0029】1. IEEE1394

まず、IEEE1394の概要について簡単に説明する。

【0030】1. 1 データ転送速度、接続トポロジー
IEEE1394 (IEEE1394-1995、IEEE1394. A) では100~400Mbpsの高速なデータ転送が可能となっている (IEEE1394. Bでは800~3200Mbps)。また、転送速度が異なるノードをバスに接続することも許される。

【0031】各ノードはツリー状に接続されており、1つのバスに最大で63個のノードが接続可能になっている。なお、バスブリッジを利用すれば約64000個のノードを接続することも可能である。

【0032】電源が投入されたり、途中でデバイスの抜き差しが発生すると、バスリセットが発生し、接続トポロジーに関する情報が全てクリアされる。そして、バスリセット後、ツリー識別 (ルートノードの決定)、自己識別が行われる。その後、アイソクロナスリソースマネージャ、サイクルマスタ、バスマネージャ等の管理ノードが決定される。そして、通常のデータ転送が開始される。

【0033】1. 2 転送方式

IEEE1394では、パケットの転送方式として、信頼性が要求されるデータの転送に好適な非同期転送と、リアルタイム性が要求される動画像や音声などのデータの転送に好適なアイソクロナス転送が用意されている。

【0034】図1 (A) に、非同期サブアクションの例を示す。1つのサブアクションは、調停、パケット転送、アクノリッジメントからなる。即ち、データ転送に先立って、まず、バスの使用权に関する調停が行われる。そしてソース (転送元) ノードからデスティネーション (転送先) ノードにパケットが転送される。このパケットのヘッダにはソースIDとデスティネーションIDが含まれる。デスティネーションノードは、このデスティネーションIDを読んで、自ノード宛のパケットか否かを判断する。デスティネーションノードは、パケットを受け取ると、ソースノードにアクノリッジメント (ACK) のパケットを返す。

【0035】パケット転送とACKの間にはアクノリッジギャップが存在する。また、1つのサブアクションと次のサブアクションの間にはサブアクションギャップが存在する。そして、サブアクションギャップに相当する一定のバス・アイドル時間が経過しないと、次のサブアクションの調停を開始できない。これによりサブアクション相互の衝突が回避される。

【0036】図1 (B) に、アイソクロナスサブアクションの例を示す。アイソクロナス転送はブロードキャスト (バスに接続される全てのノードに転送) で実行されるため、パケット受信時にACKは返送されない。また、アイソクロナス転送では、ノードIDではなくチャネル番号を使用してパケット転送が行われる。なお、サブアクション間にはアイソクロナスギャップが存在する。

【0037】図1 (C) に、データ転送時のバスの様子を示す。アイソクロナス転送は、サイクルマスタが一定周期毎にサイクルスタートパケットを発生することで開始する。これにより、1つのチャネル当たり、125μs毎に少なくとも1つのパケットを転送できるようになる。この結果、動画像や音声などのリアルタイム性が要求されるデータの転送が可能になる。

【0038】非同期転送はアイソクロナス転送の合間に行われる。即ち、アイソクロナス転送の方が非同期転送よりも優先順位が高くなっている。これは、図1 (C) に示すように、アイソクロナスギャップの時間を、非同期転送のサブアクションギャップの時間よりも短くすることで実現される。

【0039】1. 3 ツリー識別

ツリー識別はバスリセットの後に行われる。ツリー識別によりノード間の親子関係やルートノードが決定される。

【0040】まず、リーフノード (1つのノードにしか接続されていないノード) が、隣接するノードに、ペア

レントノーティファイを送る。例えば図2(A)のようにノードA、B、C、D、Eが接続されている場合には、ノードAからノードBに、ノードD及びEからノードCに、ペアレントノーティファイ(PN)が送られる。

【0041】ペアレントノーティファイを受け取ったノードは、送り元のノードを自身の子と認知する。そして、チャイルドノーティファイをそのノードに送る。例えば図2(A)では、ノードBからノードAに、ノードCからノードD及びEにチャイルドノーティファイ(CN)が送られる。これによってノードB、A間、ノードC、D間、ノードC、E間の親子関係が決定する。

【0042】ノードB、Cの親子関係は、どちらが先にペアレントノーティファイを送ったかで決定される。例えば図2(B)のように、ノードCの方が先にペアレントノーティファイを送ると、ノードBが親になりノードCが子になる。

【0043】ポートの接続先の全てのノードが自身の子となるノードがルートになる。図2(B)では、ノードBがルートになる。なお、IEEE1394では、全てのノードがルートになる可能性がある。

【0044】1.4 自己識別

ツリー識別の後、自己識別が行われる。自己識別においては、接続トポロジにおいてルートノードから遠いノードから順にセルフIDパケットが転送される。

【0045】より具体的には、例えば図3において、まず、ルートノードBのポート1(番号の小さいポート)に接続されるノードAが、セルフIDパケット(自己識別パケット)を全てのノードにブロードキャストする。

【0046】次に、ルートノードBのポート2(番号が大きいポート)に接続されるノードCが選択され、このノードCのポート1(番号の小さいポート)に接続されるノードDがセルフIDパケットをブロードキャストする。次に、ノードCのポート2(番号の大きいポート)に接続されるノードEがセルフIDパケットをブロードキャストし、その後、ノードCがブロードキャストする。最後に、ルートであるノードBがセルフIDパケットをブロードキャストし、自己識別が完了する。

【0047】セルフIDパケットには各ノードのIDが含まれる。ブロードキャストを行う時点で他のノードから受け取ったセルフIDパケットの個数が、この各ノードのIDとなる。例えば図3では、ノードAがブロードキャストを行う時点では、どのノードもセルフIDパケットを発していないため、ノードAのIDは0になる。ノードAは、このID=0をセルフIDパケットに含ませてブロードキャストする。また、ノードDがブロードキャストを行う時点では、ノードAのみがセルフIDパケットを発している。このため、ノードDのIDは1になる。同様に、ノードE、C、BのIDは、各々、2、3、4になる。

【0048】図4(A)にセルフIDパケットのフォーマットを示す。同図に示すようにセルフIDパケットには各ノードの基本情報が含まれる。具体的には、各ノードのID(PHY_ID)、リンク層がアクティブか否か(L)、ギャップカウント(gap_cnt)、転送速度(sp)、アイソクロナスリソースマネージャになれる能力を有するか否か(C)、電力状態(pwr)、ポートの状態(p0、p1、p2)などに関する情報が含まれる。

【0049】なお、図4(B)に、ノードのポート数が4個以上の場合に使用されるセルフIDパケット#1、#2、#3のフォーマットを示す。ポート数が4~11個の場合にはセルフIDパケット#0(図4(A))及び#1が、12~19個の場合にはセルフIDパケット#0、#1及び#2が、20~27個の場合にはセルフIDパケット#0、#1、#2及び#3が使用されることになる。

【0050】また、図4(C)、(D)に、セルフIDパケットと同様に、物理層のパケット(PHYパケット)であるリンクオンパケット、PHY構成パケットのフォーマットを示す。

【0051】1.5 アイソクロナスリソースマネージャ

アイソクロナスリソースマネージャ(IRM)は以下の管理機能を有する。

【0052】第1に、アイソクロナス転送に必要な種々のリソースを提供する。例えば、チャネル番号レジスタや帯域幅レジスタを提供する。第2に、バスマネージャのIDを示すレジスタを提供する。第3に、バスマネージャがない場合に、簡易的なバスマネージャとなる役割を有する。

【0053】IRMになれる能力を有し(アイソクロナスリソースを管理する能力を有し)、且つ、動作状態になっている(リンク層がアクティブになっている)ノードの中で(IRMになれる資格を有するノードの中で)、ルートに最も近い(IDが最も大きい)ノードがIRMになる。より、具体的には、図4(A)のセルフIDパケットにおいて、IRMになれる能力を有するか否かを示すC(CONTENDER)ビットと、リンク層がアクティブか否かを示すL(LINK_ACTIVE)ビットが共に1になっているノードの中で、ルートに一番近いノード(PHY_IDが一番大きいノード)がIRMになる。例えば、ルートノードのセルフIDパケットのCビットとLビットが1の場合には、ルートノードがIRMになる。

【0054】1.6 サイクルマスタ、バスマネージャ
サイクルマスタは、図1(C)に示すサイクルスタートパケットを送信する役割を有し、ルートノードがサイクルマスタになる。

【0055】バスマネージャは、トポロジマップ(各

ノードの接続状態)の作成、スピードマップの作成、バスの電力管理、サイクルマスタの決定、ギャップカウンタの最適化などの仕事を行う。

【0056】1. 7 プロトコル構成

図5を用いて、IEEE1394のプロトコル構成(層構造)について説明する。

【0057】IEEE1394のプロトコルは、物理層、リンク層、トランザクション層により構成される。また、シリアルバスマネージメントは、物理層、リンク層、トランザクション層をモニターしたり制御したりするものであり、ノードの制御やバスのリソース管理のための種々の機能を提供する。

【0058】物理層は、リンク層により使用されるロジカルシンボルを電気信号に変換したり、バスの調停を行ったり、バスの物理的インターフェースを定義する。

【0059】リンク層は、アドレッシング、データチェック、データフレーミング、サイクル制御などを提供する。

【0060】トランザクション層は、リード、ライト、ロックなどのトランザクションを行うためのプロトコルを定義する。

【0061】物理層及びリンク層は、通常、データ転送制御装置(インターフェースチップ)などのハードウェアにより実現される。また、トランザクション層は、CPU上で動作するファームウェアや、ハードウェアにより実現される。

【0062】2. 全体構成

次に、本実施形態の全体構成について図6を用いて説明する。

【0063】図6において、PHYインターフェース10は、物理層のプロトコルを実現するPHYチップとのインターフェースを行う回路である。

【0064】リンクコア20(リンク手段)は、リンク層のプロトコルやトランザクション層のプロトコルの一部を実現する回路であり、ノード間でのパケット転送のための各種サービスを提供する。レジスタ22は、これらのプロトコルを実現したりリンクコア20を制御するためのレジスタである。

【0065】FIFO(ATF)30、FIFO(ITF)32、FIFO(RF)34は、各々、非同期送信用、アイソクロナス送信用、受信用のFIFOであり、例えばレジスタや半導体メモリなどのハードウェアにより構成される。本実施形態では、これらのFIFO30、32、34の段数は非常に少ない。例えば1つのFIFOの段数は、好ましくは3段以下であり、更に好ましくは2段以下となる。

【0066】DMAC40、42、44は、各々、ATF、ITF、RF用のDMAコントローラである。これらのDMAC40、42、44を用いることで、CPU66に介入されることなく、RAM80とリンクコア20

0との間でのデータ転送が可能になる。なお、レジスタ46は、DMAC40、42、44などを制御するレジスタである。

【0067】ポートインターフェース50は、アプリケーション層のデバイス(例えばプリンタの印字処理を行うデバイス)とのインターフェースを行う回路である。本実施形態では、このポートインターフェース50を用いて、例えば8ビットのデータ転送が可能になっている。

【0068】FIFO(PF)52は、アプリケーション層のデバイスとの間でのデータ転送のためのFIFOであり、DMAC54は、PF用のDMAコントローラである。レジスタ56は、ポートインターフェース50やDMAC54を制御するレジスタである。

【0069】CPUインターフェース60は、データ転送制御装置をコントロールするCPU66とのインターフェースを行う回路である。CPUインターフェース60は、アドレスデコーダ62、データ同期化回路63、割り込みコントローラ64を含む。クロック制御回路68は、本実施形態で使用されるクロックを制御するものであり、PHYチップから送られてくるSCLKや、マスタクロックであるHCLKが入力される。

【0070】バッファマネージャ70は、RAM80とのインターフェースを管理する回路である。バッファマネージャ70は、バッファマネージャの制御のためのレジスタ72、RAM80へのバス接続を調停する調停回路74、各種の制御信号を生成するシーケンサ76を含む。

【0071】RAM80は、ランダムアクセス可能なパケット記憶手段として機能するものであり、その機能は例えばSRAM、DRAMなどにより実現される。そして、本実施形態では、図7に示すように、このRAM80がヘッダ領域(広義には制御情報領域)とデータ領域に分離されている。そして、パケットのヘッダ(広義には制御情報)は図7のヘッダ領域に格納され、パケットのデータはデータ領域に格納される。

【0072】なお本実施形態では、RAM80がデータ転送制御装置に内蔵されている。但し、RAM80の一部を外付けにすることも可能である。

【0073】バス90(或いはバス92、94)は、アプリケーションに接続されるものである(第1のバス)。またバス96(或いはバス98)はデータ転送制御装置をコントロールするためのものであり、データ転送制御装置をコントロールするデバイス(例えばCPU)に電氣的に接続される(第2のバス)。またバス100(或いはバス102、104、105、106、107、108、109)は、物理層のデバイス(例えばPHYチップ)に電氣的に接続されるものである(第3のバス)。また、バス110は、ランダムアクセス可能な記憶手段であるRAMに電氣的に接続されるものであ

る(第4のバス)。

【0074】バッファマネージャ70の調停回路74は、DMAC40、DMAC42、DMAC44、CPUインターフェース60、DMAC54からのバスアクセス要求の調停を行う。そして、この調停結果に基づいて、各々、バス105、107、109、98、94のいずれかと、RAM80のバス110との間にデータの経路が確立される(第1、第2、第3のバスのいずれかと第4のバスとの間にデータ経路が確立される)。

【0075】本実施形態の1つの特徴は、ランダムアクセスが可能でありパケットを格納するRAM80を設けると共に、互いに分離されるバス90、96、100と、これらのバスをRAM80のバス110に接続するための調停回路74とを設けた点にある。

【0076】例えば図8に、本実施形態と構成の異なるデータ転送制御装置の例を示す。このデータ転送制御装置では、リンクコア902は、PHYインターフェース900、バス922を介してPHYチップと接続される。また、リンクコア902は、FIFO904、906、908、CPUインターフェース910、バス920を介してCPU912に接続される。そして、CPU912は、バス924を介して、CPUにローカルなメモリであるRAM914に接続される。

【0077】なお、FIFO904、906、908は、図6のFIFO30、32、34と異なり、非常に段数の多いものとなる(例えば1つのFIFOが16段程度)。

【0078】図8の構成のデータ転送制御装置を用いた場合のデータ転送の手法について図9を用いて説明する。PHYチップ930を介して他のノードから送られてきた受信パケットは、バス922、データ転送制御装置932、バス920を介してCPU912が受け取る。そして、CPU912は、受け取った受信パケットをバス924を介してRAM914に書き込む。そして、CPU912は、受信パケットをアプリケーション層が使用できるように加工し、バス926を介してアプリケーション層のデバイス934に転送する。

【0079】一方、アプリケーション層のデバイス934からのデータを転送する場合には、CPU912は、このデータをRAM914に書き込む。そして、RAM914のデータにヘッダを付加することでIEEE1394に準拠したパケットを生成する。そして生成されたパケットは、データ転送制御装置932、PHYチップ930などを介して他のノードに送信される。

【0080】しかしながら、このようなデータ転送手法によると、CPU912の処理負荷が非常に重くなる。従って、ノード間を接続するシリアルバスの転送速度が高速になっても、CPU912の処理のオーバーヘッドなどに起因して、システム全体の実転送速度は低くなり、結局、高速なデータ転送を実現できない。

【0081】このような問題を解決する1つの手法として、図10に示すように、データ転送制御装置932とRAM914との間でのデータ転送や、RAM914とアプリケーション層のデバイス934との間でのデータ転送を、ハードウェアDMAにより実現する手法も考えられる。

【0082】しかしながら、この手法では、CPUバス928が、データ転送制御装置932、RAM914間でのデータ転送、RAM914、CPU912間でのデータ転送、RAM914、アプリケーション層デバイス934間でのデータ転送に使用されることになる。従って、システム全体のデータ転送の高速化を図ろうとすると、CPUバス928としてPCIバスのような高速なバスを使用しなければならなくなり、これは、データ転送制御装置を使用する電子機器の高コスト化を招く。

【0083】これに対して、本実施形態では図11に示すように、データ転送制御装置120、アプリケーション層デバイス124間のバス90と、CPUバス96と、データ転送制御装置120、RAM80間のバス110とが互いに分離されている。従って、CPUバス96をデータ転送の制御のみに使用できるようになる。また、バス90を占有して、データ転送制御装置120、アプリケーション層デバイス124間でデータ転送を行うことができるようになる。例えば、データ転送制御装置120が組み込まれる電子機器がプリンタである場合には、バス90を占有して印字データを転送できるようになる。この結果、CPU66の処理負荷を軽減でき、システム全体の実転送速度を高めることができる。またCPU66として安価なものを採用できると共に、CPUバス96として高速なバスを使用する必要性がなくなる。このため、電子機器の低コスト化、小規模化を図れるようになる。

【0084】3. パケット記憶用RAMへのパケットの格納

3.1 本実施形態の特徴

図8の比較例では、図12(A)に示すように、リンクコア902とCPU912との間には、送信用のFIFO904、906と受信用のFIFO908が設けられている。

【0085】即ち、他のノードから送られてくる受信パケットは、リンクコア902を介してFIFO908に順次入力される。そして、CPU912(CPU上で動作するファームウェアやアプリケーションソフト)は、FIFO908に入力された順序で受信パケットをFIFO908から順次読み出す。

【0086】また、CPU912は、転送すべき送信パケットをFIFO904、906に順次入力する。そして、送信パケットは、このFIFO904、906に入力された順序でリンクコア902を介して他のノードに順次転送される。

【0087】これに対して、本実施形態では、図12 (B) に示すように、ノード間でのパケット転送のための各種サービスを提供するリンクコア20と、CPU66との間に、FIFO30、32、34の他に、ランダムアクセス可能なパケット記憶手段として機能するRAM80を介在させている。

【0088】即ち、他のノードからリンクコア20を介して送られてくる受信パケットは、FIFO34に格納された後に、DMAC44（書き込み手段）によりRAM80に書き込まれる。そして、CPU66が、このランダムアクセス可能なRAM80から受信パケットを読み出す。

【0089】また、CPU66などにより、転送すべき送信パケットがRAM80に書き込まれる。そして、この書き込まれた送信パケットは、DMAC40、42（読み出し手段）により読み出され、FIFO30、32、リンクコア20を介して他のノードに転送される。

【0090】図12 (A) では、CPU912は、受信した順序でFIFO908から受信パケットを受け取らなければならない、送信する順序でFIFO904、906に送信パケットを格納しなければならない。これに対して、リンクコア20とCPU66との間にランダムアクセス可能なRAM80を介在させる本実施形態では、受信順序に依存せずに、RAM80の所望のアドレスに受信パケットを書き込むことが可能になる。また、送信順序に依存せずに、CPU66などがRAM80の所望のアドレスに送信パケットを書き込むことも可能になる。

【0091】また、図12 (A) では、FIFO904、906、908の段数を非常に多くする必要がある。これに対して、本実施形態では、FIFO30、32、34の段数を格段に減らすことができ、例えば段数を3段以下とすることも可能になる。

【0092】なお、本実施形態では、FIFO30、32、34を設けない構成にしてもよい。

【0093】また、本実施形態では、図7に示すように、RAM80の記憶領域を、パケットのヘッダ（広義には制御情報）が格納されるヘッダ領域（広義には制御情報領域）とパケットのデータが格納されるデータ領域に分離している。これは、図12 (B) に示すようにリンクコア20とCPU66との間にランダムアクセス可能なRAM80を介在させることで実現可能になる。

【0094】即ち、図12 (A) では、CPU912は、受信した順序でFIFO908から受信パケットを受け取らなければならない。従って、受信パケットをヘッダとデータに分離する処理を実現しようとする、FIFO908から読み出した受信パケットをCPU912がローカルメモリであるRAMに一旦書き込み、そのRAMからCPU912が受信パケットを読み出してヘッダとデータに分離する必要がある。また、図12

(A) では、CPU912は、送信する順序でFIFO904、906に送信パケットを入力しなければならない。例えば、パケット1（ヘッダ1、データ1）、パケット2（ヘッダ2、データ2）、パケット3（ヘッダ3、データ3）を送信する場合には、ヘッダ1、データ1、ヘッダ2、データ2、ヘッダ3、データ3というような順序でFIFO904、906に送信パケットを入力しなければならない。従って、CPU912による並べ替え処理が必要になる。

【0095】このように図12 (A) では、CPU912の処理負荷が非常に重くなり、これは結局、システム全体の実転送速度の低下という事態を招く。

【0096】これに対して、本実施形態では、RAM80の記憶領域がヘッダ領域とデータ領域に分離される。より具体的には、図13に示すように、各受信パケットのヘッダとデータがハードウェアにより分離され、ヘッダがヘッダ領域に、データがデータ領域に格納される。また、図14に示すように、ヘッダ領域に格納されたヘッダとデータ領域に格納されたデータとがハードウェアにより結合され、各ノードに転送すべき送信パケットが組み立てられる。従って、CPU66の処理負荷が図12 (A) に比べて非常に軽くなり、システム全体の実転送速度を向上できる。また、CPU66として安価なものを採用できると共にCPU66に接続されるバスも低速なもので済むため、データ転送制御装置や電子機器の小規模化、低コスト化を図れるようになる。

【0097】また、本実施形態によれば、ヘッダはヘッダ領域にまとめられて格納され、データもデータ領域にまとめられて格納される。従って、ヘッダやデータの読み出し処理や書き込み処理の簡易化を図ることができ、処理のオーバーヘッドを軽減化できる。例えば、図11の手法でデータ転送を行う場合には、CPU66は、CPUバス96を介してヘッダ領域にのみアクセスしヘッダを読み出したり書き込んだりするだけでデータ転送を制御できるようになる。また、アプリケーション層のデバイス124は、データ領域のデータを連続してバス90を介して読み出ししたり、データを連続してデータ領域に書き込んだりすることが可能になる。

【0098】なお、図15に示すように、ヘッダ領域に格納される各ヘッダと、データ領域に格納される各データとは、ヘッダに含ませたデータポイントにより対応させることが望ましい。この場合、受信パケットのヘッダへのデータポイントの付加は、例えばリンクコア20が行い、送信パケットのヘッダへのデータポイントの付加は、例えばファームウェアが行うようにする。

【0099】また、本実施形態では、図16に示すように、受信パケットのスタート（ヘッダの最初）、ヘッダ、データ、トレーラーを区別するためのTAGを生成し、このTAGを受信パケットに関連づけている。より具体的には、図6のリンクコア20が、受信パケットの

スタート、ヘッダ、データ、トレイラーをFIFO34に転送する際に、図16に示す2ビットのTAGも同時にFIFO34に転送している。そして、本実施形態では、受信パケットに関連づけられたこのTAGを利用して、図13に示すように受信パケットをヘッダとデータに分離しヘッダ領域とデータ領域に格納している。より具体的には、図6のDMAC44が、受信パケットと共にFIFO34から出力されるTAGを利用して、受信パケットをヘッダとデータに分離し、RAM80に書き込んでいる。なお、TAGは、少なくともヘッダとデータを区別できるものであればよい。

【0100】例えば、TAGを利用せずに受信パケットのヘッダとデータを分離する手法として、ヘッダに含まれるtcode（トランザクションコード）を利用する手法が考えられる。即ち、tcodeをデコードして、ヘッダのサイズを調べ、そのサイズの分だけ受信パケットをヘッダ領域に格納し、残りをデータ領域に格納する。

【0101】しかしながら、この手法では、tcodeをデコードする回路が必要となり、回路が大規模化してしまう。

【0102】これに対して、TAGを利用すれば、DMAC44は、TAGを見るだけで受信パケットをヘッダとデータに分離できる。従って、回路をそれほど大規模化することなく、簡易な処理で受信パケットのヘッダとデータを分離できるようになる。

【0103】また、本実施形態では、図17(A)に示すように、ヘッダ領域とデータ領域の他に、これらの領域と分離されたCPU66のワーク領域をRAM80に設けている。このようにすれば、CPU66が、ワーク領域に直接アクセスして、CPU66の処理のためにその領域を利用できるようになる。従って、CPU66がローカルメモリを有する場合には、そのメモリを小容量化することが可能になる。更に、CPU66のローカルメモリを設ける必要性自体を無くすことも可能になる。これにより、データ転送制御装置や電子機器の小規模化、低コスト化を図れるようになる。

【0104】また、本実施形態では、図17(B)に示すように、ヘッダ領域を受信用と送信用の領域に分離（分割）している。これらの受信用、送信用のヘッダ領域の各々は、少なくとも1個以上のパケットを格納できるものであればよい。

【0105】例えば図8の構成では、ローカルメモリであるRAM914にCPU912が受信パケットや送信パケットを格納する。このため、CPU912は、受信パケットや送信パケットを格納したアドレスを各パケット毎に保持し管理しなければならない。従って、CPU912の処理が煩雑化し、処理負担が増加する。特に、CD-Rドライブのように受信処理と送信処理が混在するアプリケーションでは、受信パケットと送信パケット

が混在してRAM914に格納されることになるため、処理の煩雑化、処理負担の増加の問題は更に深刻になる。

【0106】これに対して、本実施形態によれば、ヘッダ領域を受信用と送信用の領域に分離される。従って、RAM80の受信用ヘッダ領域には受信用のヘッダが、送信用ヘッダ領域には送信用のヘッダが規則的に格納される。従って、CPU66は、受信処理の際には受信用ヘッダ領域にアクセスし、送信処理の際には送信用ヘッダ領域にアクセスすればよいことになる。そして、例えば受信パケットを連続して受信する場合には、受信用ヘッダ領域からヘッダを連続して読み出し、送信パケットを連続して送信する場合には、送信用ヘッダ領域にヘッダを連続して書き込めばよいことになる。従って、CPU66の処理負担を大幅に軽減でき、システム全体の実転送速度の向上を期待できるようになる。

【0107】また、本実施形態では、図17(C)に示すように、データ領域を受信用と送信用の領域に分離している。

【0108】このようにすれば、例えば図11において、アプリケーション層のデバイス124は、RAM80の受信用データ領域から連続して受信データを読み出せるようになると共に、送信用データ領域に連続して送信データを書き込めるようになる。そして、読み出しや書き込みの際にはアドレスを単純にインクリメント（或いはデクリメント）すればよいので、アドレス制御も容易になる。この結果、本実施形態によれば、処理のオーバーヘッドを格段に軽減でき、実転送速度の向上を期待できるようになる。

【0109】また、本実施形態では、データ領域を、アイソクロナス転送用と非同期転送用の領域に分離している。より具体的には例えば図17(D)に示すように、データ領域を、アイソクロナス受信用、非同期受信用、アイソクロナス送信用、非同期送信用の領域に分離している。

【0110】例えば図8の構成では、アイソクロナスパケットと非同期パケットとが混在してFIFO908に入力される。そして、CPU912は、この入力された順序でFIFO908からパケットを読み出す。また、この読み出した順序でパケットをローカルメモリであるRAM914に格納する。従って、RAM914には、アイソクロナスパケットと非同期パケットが混在して格納されることになる。

【0111】さて、アイソクロナスパケットの処理にはリアルタイム性が要求される。従って、CPU912は一定期間内にアイソクロナスパケットに関する処理を終了しなければならない。ところが、図8では、CPU912はFIFO908に入力された順序でFIFO908からパケットを読み出さなければならない。また、RAM914には、アイソクロナスパケットと非同期パケッ

トが混在するようになる。従って、例えば非同期パケットの次にアイソクロナスパケットが送られてきた場合にも、非同期パケットを先に処理しなければならず、アイソクロナスパケットを先に処理することができない。このため、リアルタイム性が要求されるアイソクロナスパケットの処理が間に合わなくなる可能性がある。

【0112】これに対して、本実施形態では、アイソクロナスパケットはアイソクロナス転送用の領域に格納され、非同期パケットは非同期転送用の領域に格納される。従って、非同期パケットの次にアイソクロナスパケットが送られてきた場合にも、アイソクロナスパケットを先に読み出し、先に処理することが可能になる。この結果、アイソクロナスパケットの処理を一定期間内に完了でき、アイソクロナスパケットの処理に要求されるリアルタイム性を維持できるようになる。

【0113】また、データ領域をアイソクロナス転送用と非同期転送用の領域に分離することで、アプリケーション層のデバイスの処理も簡易化できる。例えばデジタルビデオカメラにおいて、リアルタイム性が要求される動画像をアイソクロナスで転送し、信頼性が要求される静止画像を非同期で転送する場合を考える。そして、図11において、データ転送制御装置120とアプリケーション層のデバイス124との間でバス90を用いて直接に動画像や静止画像のデータを転送したとする。この場合に、データ領域をアイソクロナス転送用と非同期転送用の領域に分離することにより、バス90を用いた動画像や静止画像のデータの転送を簡易化できる。動画像のデータを転送する場合には、動画像のデータをまとめてアイソクロナス転送用のデータ領域に書き込み、静止画像のデータを転送する場合には、静止画像のデータをまとめて非同期転送用のデータ領域に書き込めばよいからである。また、この書き込みの際には、アドレスを単純にインクリメント（或いはデクリメント）すればよいので、アドレス制御も容易になるからである。

【0114】また、本実施形態では、非同期転送用データ領域を、複数の領域に分離している。より具体的には例えば図17(E)に示すように、非同期受信信用領域を、第1、第2の非同期受信信用の領域に分離している。また、非同期送信信用領域を、第1、第2の非同期送信信用の領域に分離している。

【0115】例えば、プリンタなどのアプリケーションでは、コマンドデータ、ステータスデータなどの制御用データのみならず、印字データも非同期転送される。この場合に、非同期転送用データ領域が複数の領域に分離されていないと、制御用データと印字データとが、非同期転送用データ領域に混在して格納されてしまう。従って、印字データの連続性を維持できなくなり、上層の処理負荷が重くなってしまう。

【0116】図17(E)に示すように、非同期転送用データ領域を複数の領域に分割すれば、例えば第1の非

同期転送用データ領域には、コマンドデータ、ステータスデータなどの制御用データを格納し、第2の非同期転送用データ領域には、印字データを格納することが可能になる。この結果、第2の非同期転送用データ領域に印字データを連続して格納することが可能になり、上層の処理負担を軽減できる。

【0117】また、本実施形態では、データ領域を、アイソクロナス転送用データ領域と第1の非同期転送用データ領域に分離する共に、アイソクロナス転送用データ領域を、随時、第2の非同期転送用データ領域としても使用している。より具体的には例えば図17(F)に示すように、アイソクロナス受信信用領域を第2の非同期受信信用領域として随時使用したり、アイソクロナス送信信用領域を第2の非同期送信信用領域として随時使用したりする。

【0118】例えば、デジタルビデオなどのアプリケーションでは、画像データ（特に動画像データ）はアイソクロナス転送される一方で、コマンドデータ、ステータスデータなどの制御用データは非同期転送される。従って、この場合には、画像データをアイソクロナス転送用データ領域に格納するようにすると共に、制御用データを第1の非同期転送用データ領域に格納するようにする。

【0119】一方、プリンタなどのアプリケーションでは、印字データも、コマンドデータ、ステータスデータなどの制御用データも、共に非同期転送される。従って、この場合には、図17(F)に示すように、アイソクロナス転送用データ領域を第2の非同期転送用データ領域として使用し、印字データをこの第2の非同期転送用データ領域に格納するようにする。一方、制御用データについては、第1の非同期転送用データ領域に格納するようにする。

【0120】このように、アプリケーションに応じて、アイソクロナス転送用データ領域を第2の非同期転送用データ領域として随時使用することで、限られたリソースの有効利用を図ることができる。

【0121】また、本実施形態では、RAM80が複数の領域に分離される場合において、各領域のサイズを可変に制御するようにしている。より具体的には図18に示すように、各領域の境界のアドレスを指すポインタP1～P6を可変に制御する。

【0122】このようにすれば、アプリケーションに応じた最適な領域分割を実現できる。例えば、プリンタのように、パケットを送信することが少なく、パケットを受信することが多いアプリケーションでは、受信信用の領域のサイズを大きくする。また、スキヤナのように、パケットを受信することが少なく、パケットを送信することが多いアプリケーションでは、送信信用の領域のサイズを大きくする。また、非同期転送を行うことが多いアプリケーションでは非同期転送用の領域のサイズを大きく

し、アイソクロナス転送を行うことが多いアプリケーションではアイソクロナス転送用の領域のサイズを大きくする。或いは、非同期転送のみを行うアプリケーションでは、アイソクロナス転送用の領域を零にする。このようにすることにより、限られたリソース（少ない記憶容量のRAM80）を有効利用することが可能になる。

【0123】また、図8の構成でFIFO904、906、908の各々のサイズを可変にする手法には、ハードウェアの構成が複雑になるという問題点がある。これに対して、RAM80の各領域のサイズを可変制御するためには、ポインタP1～P6の指すアドレスを制御する（ポインタの指すアドレスが格納されるレジスタの内容を制御する）のみでよい。このため、簡易なハードウェア構成で各領域のサイズを可変に制御できる。

【0124】なお、RAM80の各領域のサイズは、電源投入後もダイナミックに可変制御できることが望ましい。例えば、CD-Rドライブのようなアプリケーションでは、CD-Rへのデータの書き込み時には受信処理が主に行われ、CD-Rからのデータの読み出し時には送信処理が主に行われる。従って、CD-Rへのデータの書き込み時には、受信用の領域のサイズが大きくなるようにポインタをダイナミックに切り替える。また、CD-Rからのデータの読み出し時には、送信用の領域のサイズが大きくなるようにポインタをダイナミックに切り替える。このようにすることで、限られたリソースを有効利用することが可能になる。

【0125】なお、本実施形態において分離された各領域は、リングバッファ構造とすることが望ましい。即ち図19(A)に示すように、RAM80の例えば受信用ヘッダ領域の一方の境界（上側の境界）から他方の境界（下側の境界）に向かってパケットのヘッダを格納する。そして、図19(B)に示すように他方の境界に達した場合には、図19(C)に示すように一方の境界に戻ってヘッダを格納するようにする（上書きする）。このようにすることで、受信ヘッダ領域には、受信した順序でヘッダが格納されるようになるため、どのような順序でパケットが受信されたかをファームウェアやアプリケーションソフトが知ることができるようになる。即ち、ランダムアクセス可能なRAM80を使用しながら、FIFOが持つ有利な機能も得ることができる。また、他方の境界に達した場合に一方の境界からパケットのヘッダを格納するようにすることで、限られたリソース（少ない記憶容量のRAM80）を有効利用できるようになる。

【0126】なお、送信用のヘッダ領域や、データ領域についても、このようなリングバッファ構造にしてもよい。

【0127】なお、本実施形態では、1つのパケットのヘッダ（広義には制御情報）やデータが格納される領域（以下、格納領域と呼ぶ）のサイズが、パケットのヘッ

ダやデータのサイズに依存せずに固定となるようにすることが望ましい。例えば図20(A)では、各ヘッダのサイズは互いに異なっているが、格納領域のサイズは固定になっている。この場合に、パケットのヘッダのサイズに依らずに、格納領域のサイズを例えば8クワドレット単位にする。このようにすれば、ファームウェアのアドレス制御などを簡易化することができ、処理のオーバーヘッドを軽減できる。特に、図19(A)、(B)、(C)のようなリングバッファ構造の場合には、格納領域のサイズを固定にしないと、図20(B)に示すように、ヘッダNが下側と上側に分離されて格納されてしまう。これは、ファームウェアの処理負担の増大化という事態を招く。格納領域のサイズを固定にすれば、このような事態を解消できる。

【0128】3.2 受信側の構成

次に受信側の構成について説明する。図21に、リンクコア20、FIFO34、DMAC44の詳細な構成の一例を示す。

【0129】リンクコア20は、バス監視回路130、直列・並列変換回路132、パケット整形回路160を含む。そして、パケット整形回路160は、パケット診断回路142、シーケンサ167、バッファ168、セクタ170を含み、パケット診断回路142は、TAG生成回路162、ヘッダ&トレイラー生成回路164、エラーチェック回路166を含む。

【0130】ここで、バス監視回路130は、PHYインターフェース10を介してPHYチップに接続される8ビット幅のデータバスD、2ビット幅のコントロールバスCTLを監視する回路である。

【0131】直列・並列変換回路132は、データバスDのデータを32ビットのデータに変換する回路である。例えば、転送速度が400Mbpsの場合には8ビットのデータが32ビットのデータに、200Mbpsの場合には4ビットのデータが32ビットのデータに、100Mbpsの場合には2ビットのデータが32ビットのデータに変換される。

【0132】パケット診断回路142は、パケットの診断を行う回路である。TAG生成回路162は、ヘッダ、データ、トレイラーなどを区別するためのTAGを生成する回路であり、ヘッダ&トレイラー生成回路164は、ヘッダ及びトレイラー（フッター）を生成する回路である。また、エラーチェック回路166は、パケットに含まれるパリティなどのエラーチェック情報をチェックしてエラーを検出する回路である。

【0133】シーケンサ167は各種の制御信号を生成するものである。バッファ168、セクタ170は、直列・並列変換回路132からのDI、パケット診断回路142からのヘッダ及びトレイラー、DMAC44からのデータポインタのいずれかを、パケット診断回路142からの信号SELにより選択するためのものであ

る。

【0134】FIFO34は、リンクコア20からの出力データであるRDの位相と、RAM80への書き込みデータであるWDATAの位相とを調整するためのバッファとして機能するものであり、FIFO状態判断回路35を含む。FIFO状態判断回路35は、FIFOが空になると、EMPTYをアクティブにし、FIFOがフルになると、FULLをアクティブにする。

【0135】DMAC44は、パケット分離回路180、アクセス要求実行回路190、アクセス要求発生回路192を含む。

【0136】パケット分離回路180は、パケット整形回路160により整形されたパケットを分離して、ヘッダ及びトレイラーをRAM80のヘッダ領域に、データをデータ領域に書き込むための回路である(図13参照)。パケット分離回路180は、TAG判別回路182、ポインタ更新回路184、アドレス発生回路188を含む。

【0137】TAG判別回路182は、TAG生成回路162により生成されたTAG(DTAG)を判別する回路である。

【0138】ポインタ更新回路184は、TAG判別回路182の出力を受け、RAM80にヘッダやデータを書き込むためのヘッダポインタやデータポインタを更新するための回路である。

【0139】アドレス発生回路188は、ポインタ更新回路184の出力を受け、RAM80への書き込みアドレスWADRを発生する回路である。

【0140】アクセス要求実行回路190は、リンクコア20からのアクセス要求を実行するための回路である。アクセス要求実行回路190は、FIFO状態判断回路35からのFULLがアクティブになると、FULLをアクティブにする。パケット整形回路160内のシーケンサ167は、FULLがアクティブでないことを条件に、RD(RxData)のストロブ信号であるRDSをアクティブにする。

【0141】なおRFAILは、受信における失敗を、シーケンサ167がアクセス要求実行回路190に対して知らせるための信号である。

【0142】アクセス要求発生回路192は、RAM80へのアクセス要求を発生するための回路である。アクセス要求発生回路192は、バッファマネージャ70からの書き込みアクロリッジメントであるWACKやFIFO状態判断回路35からのEMPTYを受け、書き込み要求であるWREQをバッファマネージャ70に出力する。

【0143】3.3 受信側の動作

次に、受信側の動作の詳細について図22のタイミング波形図などを用いて説明する。

【0144】まず、リンクコア20の動作について説明

する。

【0145】PHYチップを介して他のノードからのパケットを受信すると、パケット診断回路142がそのパケットを診断する。そして、ヘッダ&トレイラー生成回路164がヘッダを生成(整形)する。このヘッダは、バッファ168を介してセクタ170に入力され、パケット診断回路142からの信号SELに基づきセクタ170がこのヘッダを選択する。これにより、図22のA1に示すように、RDとしてヘッダ(H0~H4)がFIFO34に出力されることになる。

【0146】なお、図23(A)に、シリアルバス上で転送される非同期パケットのフォーマット(IEEE1394規格)を示す。一方、図23(B)に、RAM80のヘッダ領域に格納される非同期受信パケットのヘッダ部分のフォーマットを示す(同図において網掛けとなっている部分がトレイラーである)。このように本実施形態では、図23(A)に示すフォーマットのパケットを、ファームウェアなどの上層が使用できるように、図23(B)に示すフォーマットのパケットに整形している。

【0147】また本実施形態では、ヘッダの第4クワドレットであるH4(図22のA2)は、図23(B)に示すようにデータ領域からデータを取り出すためのデータポインタになっている。このデータポインタ(H4)は、DMAC44(ポインタ更新回路184)からバッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。このように、パケット整形回路160は、DMAC44からデータポインタを受け取り、そのデータポインタを、RAM80に書き込まれるヘッダに埋め込んでいる。

【0148】次に、PHYチップから、データバスDを介してパケットのデータ部分が送られてくる。直列・並列変換回路132は、このデータ部分を32ビットのデータであるDIに変換し、パケット診断回路142及びバッファ168に出力する。

【0149】なお、DIEは、DIが有効か無効かを示す信号であり、DISは、DIの取り込みタイミングを知らせるためのストロブ信号である。

【0150】直列・並列変換回路132からのDIは、バッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。これにより、A3に示すように、RDとしてデータD0~DnがFIFO34に出力される。

【0151】次に、ヘッダ&トレイラー生成回路164からのトレイラーがバッファ168を介してセクタ170に入力され、セクタ170がこれを選択する。これにより、A4に示すように、RDとしてトレイラー(H5。図23(B)の網掛け部分)がFIFO34に出力される。

【0152】さて、TAG生成回路162は、RDとし

て出力される情報を区別するためのTAGを生成している。本実施形態では図16に示すようにTAGは2ビットであり、(00)、(01)、(10)、(11)は、各々、ヘッダ、トレイラー、データ、スタート(ヘッダの最初)を表す。従って、例えば図22では、(11)、(00)、…、(10)、…、(01)というようにTAGが変化する。FIFO34には、この2ビットのTAGと32ビットのRDとからなる34ビットのデータが入力されることになる。

【0153】次に、FIFO34の動作について説明する。

【0154】FIFO34は、リンクコア20からのTAG、RDを受け、A5、A6に示すように、DTAG、WDATAとして出力する。

【0155】FIFO34内のFIFO状態判断回路35は、内蔵するカウンタにより、FIFO34のデータ数(FIFOカウント)をカウントする。そして、FIFO34がフル(データ数=2)になった場合には、図22のA7に示すようにFULLをアクティブ(Hレベル)にする。また、FIFO34が空(データ数=0)になった場合には、A8に示すようにEMPTYをアクティブにする。FIFO34がフルになったことは、FULL、FFULLにより、DMAC44内のアクセス要求実行回路190やリンクコア20内のシーケンサ167に伝えられる。また、FIFO34が空になったことは、EMPTYにより、DMAC44内のアクセス要求発生回路192に伝えられる。

【0156】次に、DMAC44の動作について説明する。

【0157】アクセス要求発生回路192は、A9に示すようにEMPTYが非アクティブ(Lレベル)になったこと(FIFO34が空でないこと)を条件に、A10に示すようにWREQをアクティブにする。そして、バッファマネージャ70からWACKを受け取ると、WREQを非アクティブにする。

【0158】さて、本実施形態では、受信時のバス調停においては、DMAC44からのアクセス要求の優先順位を最も高くしている。従って、A10、A11に示すように、DMAC44からのWREQと、CPUインターフェース60やポート用のDMAC54からのOtherWREQとが競合した場合には、WREQの方が優先する。即ち、A12、A13に示すように、WACKの方がOtherWACKよりも先にアクティブになる。このように、WREQとOtherWREQが競合した場合に、WREQの方を優先させるのは、以下の理由による。即ち、IEEE1394では、SCSIなどとは異なり、他のノードからのパケットが転送クロックに同期して次々に転送されてくる。従って、これらの間断なく転送されてくるパケットを優先してRAM80に次々に格納する必要があるからである。

【0159】また、本実施形態では、CPUインターフェース60やポート用のDMAC54からのアクセス要求をバッファマネージャ70が受け付けている間は、A14に示すように、DMAC44のアクセス要求は所与の期間だけ待たされる。従って、リンクコア20からのRDと、バッファマネージャ70へのWDATAとは同期しない。この理由により、本実施形態では、RDとWDATAの位相を調整するためのFIFO34を設けている。この場合、FIFO34は、位相調整のために必要な最低限の段数(好ましくは3段以下、更に好ましくは2段以下)を備えていれよい。

【0160】パケット分離回路180に含まれるTAG判別回路182は、WDATAと共にFIFO34から出力されるDTAGを判別し、WDATAが、スタート(ヘッダの最初)、ヘッダ、データ、トレイラーのいずれなのかを判定する。そして、ポインタ更新回路184は、この判定結果に基づいて、ヘッダポインタやデータポインタの更新を行う。次に、アドレス発生回路188は、更新されたヘッダポインタやデータポインタに基づいて、WDATAの書き込みアドレスであるWADRを発生する。

【0161】より具体的には、例えば、WDATAがスタート又はヘッダであるとDTAGに基づき判定された場合は、ポインタ更新回路184が、図24(A)に示すように、ヘッダポインタHPのインクリメント(広義には更新)を行う。そして、アドレス発生回路188は、図22のA15に示すように、インクリメントされるヘッダポインタに応じたWADRを発生する。

【0162】次に、WDATAがデータであるとDTAGに基づき判定された場合は、ポインタ更新回路184が、図24(B)に示すように、データポインタDPのインクリメントを行う。このデータポインタDPは、パケット整形回路160によりヘッダの第4クワドレットに埋め込まれたH4に相当する。アドレス発生回路188は、図22のA16に示すように、インクリメントされるデータポインタに応じたWADRを発生する。

【0163】次に、WDATAがトレイラーであるとDTAGに基づき判定された場合は、ポインタ更新回路184が、図24(C)に示すように、ヘッダポインタのインクリメントを行う。そして、アドレス発生回路188は、図22のA17に示すように、インクリメントされるヘッダポインタに応じたWADRを発生する。

【0164】なお、最終的には図24(D)に示すように、ヘッダポインタは、処理対象となったパケットのヘッダ部分の下側の境界(次のパケットのヘッダ部分の上側の境界)を指すようになる。また、データポインタは、パケットのデータ部分の下側の境界(次のパケットのデータ部分の上側の境界)を指すようになる。ヘッダポインタやデータポインタの最終位置は、受信に失敗がなかった(RFAILが非アクティブ)ことを条件に、

図6のレジスタ46内のヘッダポインタ設定レジスタやデータポインタ設定レジスタにリストアされる。

【0165】以上のようにして、パケットを分離してヘッダ領域とデータ領域に書き込むことが可能になる。

【0166】特に本実施形態では、ヘッダに付加されるデータポインタが、ポインタ更新回路184からパケット整形回路160に伝えられる。そしてパケット整形回路160が、この伝えられたデータポインタをパケットのヘッダに付加する。このようにすることで、ヘッダ領域からヘッダを読み出したファームウェアなどが、そのヘッダに対応するデータのデータ領域での格納アドレスを容易に知ることができるようになる。また、データポインタの付加は、パケット整形回路160により行われ、DMAC44はこれに関与する必要がない。従って、DMAC44がRAM80へのデータ書き込み処理に専念できるようになり、DMAC44の回路構成や処理を簡素化できるようになる。

【0167】なお、ヘッダ領域とデータ領域の境界などの、RAM80を分離する領域の境界(図18のP1~P6)の設定は、CPUインターフェース60を介してCPU66(ファームウェア等)が、図6のレジスタ46に含まれるポインタ設定レジスタに対して、境界のアドレスを指すポインタを設定することで実現される。

【0168】また、図17(D)、(E)、(F)に示すように、データ領域が複数の領域に分離される場合には、例えばtcodeなどのパケットの制御情報に基づき、分離された複数の領域のいずれかにパケットのデータを書き込むようにすることが望ましい。

【0169】より具体的には、図21に示すように、DMAC44が、複数のデータポインタ、例えば第1、第2のデータポインタをパケット整形回路160に対して渡すようにする(3個以上のデータポインタを渡してもよい)。そして、パケット整形回路160は、例えば、アイソクロナス転送時(或いは図17(F)の第2の非同期転送時)にはDMAC44からの第1のデータポインタを選択し、非同期転送時(或いは図17(F)の第1の非同期転送時)にはDMAC44からの第2のデータポインタを選択するようにする。即ち、パケット整形回路160内のパケット診断回路142が、tcodeなどのパケットの制御情報に基づいて、アイソクロナス転送か非同期転送か(或いは第2の非同期転送か第1の非同期転送か)を判別し、この判別結果に基づき信号SELを制御する。そして、バッファ168を介してセレクタ170に入力される第1、第2のデータポインタのいずれかを選択するようにする。これにより、アイソクロナス転送(或いは第2の非同期転送)のパケットには第1のデータポインタが埋め込まれ、非同期転送(或いは第1の非同期転送)のパケットには第2のデータポインタが埋め込まれるようになる。この結果、データ領域を分離する特定の領域に、連続的にデータを格納すること

が可能になる。即ちデジタルカメラにおける動画像データをアイソクロナス転送用データ領域に連続的に格納したり、プリンタにおける印字データを第2の非同期転送用データ領域に連続的に格納したりすることが可能になる。

【0170】なお、図17(E)のように、データ領域をアイソクロナス転送用データ領域と非同期転送用データ領域に分離すると共に、非同期転送用データ領域を第1、第2の非同期転送用データ領域に分離する場合には、3つのデータポインタを用意することが望ましい。即ち、アイソクロナス転送用データ領域を指す第1のデータポインタと、第1、第2の非同期転送用データ領域を指す第2、第3のデータポインタを用意するようにする。このようにすることで、図17(E)に示すようなデータ領域の分離を、容易に実現できるようになる。

【0171】3. 4 送信側の構成

次に送信側の構成について説明する。図25に、FIFO30、DMAC40の詳細な構成の一例を示す。

【0172】位相調整用のバッファとして機能するFIFO30は、FIFO状態判断回路31を含む。FIFO状態判断回路31は、FIFOが空になると、EMPTYをアクティブにし、FIFOがフルになると、FULLをアクティブにする。

【0173】DMAC40は、パケット結合回路280、アクセス要求実行回路290、アクセス要求発生回路292、ACK書き込み要求発生回路294、ACK書き込みデータ&アドレス発生回路296を含む。

【0174】パケット結合回路280は、ヘッダをRAM80のヘッダ領域から読み出し、データをデータ領域から読み出し、これらのヘッダとデータによりフレームが構成される送信パケットを組み立てる回路である(図14参照)。パケット結合回路280は、ポインタ更新回路284、アドレス発生回路288を含む。

【0175】ポインタ更新回路284は、RAM80からヘッダやデータを読み出すためのヘッダポインタやデータポインタを更新するための回路であり、データポインタ取得回路285を含む。データポインタ取得回路285は、RAM80から読み出されたRDAT Aからデータポインタを取得する回路であり、tcode判別回路286を含む。

【0176】アドレス発生回路288は、ポインタ更新回路284の出力などを受け、RAM80の読み出しアドレスRADRを発生する回路である。

【0177】アクセス要求実行回路290は、FIFO状態判断回路31からのEMPTYがアクティブになると、FIFOINをアクティブにする。リンクコア20は、FIFOINがアクティブでないことを条件に、TD(TxData)のストロブ信号であるTDSをアクティブにする。

【0178】なおTFAILは、送信における失敗を、

リンクコア20がアクセス要求実行回路290に対して知らせるための信号である。

【0179】アクセス要求発生回路292は、バッファマネージャ70からの読み出しアクノリッジメントであるRACKやFIFO状態判断回路31からのFULLを受け、読み出し要求であるRREQをバッファマネージャ70に出力する。

【0180】ACK書き込み要求発生回路294は、リンクコア20からのTCMPやバッファマネージャ70からのWACKを受け、WREQをバッファマネージャ70に出力する。また、ACK書き込みデータ&アドレス発生回路296は、リンクコア20からのTACKを受け、送信パケットに書き戻すACKのコードをWDATAとして出力し、ACKを書き戻すアドレスをWADDRとして出力する。

【0181】3. 5 送信側の動作

次に、送信側の動作の詳細について図26のタイミング波形図などを用いて説明する。

【0182】まず、リンクコア20の動作について説明する。

【0183】送信開始を知らせるTSTARTがアクティブになると、図26のB1に示すように、リンクコア20は、ストロブ信号であるTDSを用いてFIFO30からTDを取り込む。この場合、リンクコア20には、ヘッダ(H0~H3)、データ(D0~Dn)の順でTDSが取り込まれる。

【0184】なお、図27に、RAM80のヘッダ領域に格納される非同期送信パケットのヘッダ部分のフォーマットを示す。同図に示すようにヘッダの第4クワドレットはデータポインタになっている。

【0185】さて、図26のB2に示す位置では、リンクコア20は、TDSをアクティブにしない。従って、B3に示すように、ヘッダの第4クワドレットであるH4はリンクコア20に取り込まれない。図27に示すように第4クワドレットのH4はデータポインタであり、リンクコア20は、このデータポインタを必要としないからである。そして、リンクコア20は、B3に示す期間において、ヘッダCRC(図23(A)参照)を生成しヘッダに付加する処理を行っている。

【0186】1つのパケットの送信処理が終了すると、リンクコア20は、B4に示すようにTCMPをアクティブにする。そして、送信のデスティネーションノードからPHYチップを介して返されてきたACKのコードを(図1(A)参照)、B5に示すようにTACKとしてDMAC40に出力する。このACKのコードは、ACK書き込み要求発生回路294、ACK書き込みデータ&アドレス発生回路296により、RAM80のヘッダ領域に格納されているヘッダに書き戻されることになる(図27の第7クワドレット)。

【0187】次に、FIFO30の動作について説明す

る。

【0188】FIFO30は、バッファマネージャ70からのRDATAを受け、TDとしてリンクコア20に出力する。

【0189】FIFO30内のFIFO状態判断回路31は、内蔵するカウンタにより、FIFO30のデータ数(FIFOカウンタ)をカウントする。そして、FIFO30が空(データ数=0)になった場合には、図26のB6に示すようにEMPTYをアクティブにする。また、FIFO30がフル(データ数=2)になった場合には、B7に示すようにFULLをアクティブ(Hレベル)にする。FIFO30が空になったことは、EMPTY、FIFOINにより、DMAC40内のアクセス要求実行回路290やリンクコア20に伝えられる。また、FIFO30がフルになったことは、FULLにより、DMAC40内のアクセス要求発生回路292に伝えられる。

【0190】次に、DMAC40の動作について説明する。

【0191】アクセス要求発生回路292は、B8に示すように、FULLが非アクティブ(Lレベル)であること(FIFO34がフルでないこと)を条件にRREQをアクティブにする。そして、バッファマネージャ70からRACKを受け取ると、RREQを非アクティブにする。

【0192】なお、本実施形態では、送信時のバス調停においては、DMAC40(或いはDMAC42)からのアクセス要求の優先順位を最も高くしている。従って、DMAC40からのRREQと、CPUインターフェース60やポート用のDMAC54からのアクセス要求(OtherRREQ)とが競合した場合には、RREQの方が優先する。一方、RREQよりも先に、CPUインターフェース60やポート用のDMAC54からのアクセス要求があった場合には、B9に示すように、DMAC40のアクセス要求は所与の期間だけ待たされる。従って、バッファマネージャ70からのRDATAと、リンクコア20へのTDとは同期しない。この理由により、本実施形態では、RDATAとTDの位相を調整するためのFIFO30を設けている。

【0193】さて、送信が開始すると、ポインタ更新回路284が、ヘッダポインタHPのインクリメント(広義には更新)を行う。そして、アドレス発生回路288は、図26のB10に示すように、インクリメントされるヘッダポインタに応じたRADRを発生する。このようにして、RDATAのヘッダ部分がRAM80から順次読み出される。

【0194】RDATAとしてH4が読み出されると、パケット結合回路280に含まれるデータポインタ取得回路285が、このH4を、データポインタDPとして取得する。より具体的には、RDATAとしてH0が読

み出されると、データポインタ取得回路285内のtc
ode判別回路286が、H0に含まれるtc
ode
(図27参照)を判別する。そして、ヘッダの例えば第
4クワドレットにデータポインタがあるとtc
odeに
基づき判断された場合には、RDATAとしてH4が読
み出された時に、データポインタ取得回路285がこの
H4を取得する。即ち、図26のB11に示すように、
RDATAのH4が、データポインタとして取得され、
RADRとして出力される。

【0195】なお、本実施形態では、B3、B11に示
すように、リンクコア20がヘッダCRCを生成してい
る期間を利用して、RDATAからデータポインタであ
るH4を取得している。即ち、本実施形態では、ヘッダ
CRCの生成はリンクコア20が行い、DMAC40は
これに関与しない。一方、データポインタの取得はDM
AC40が行い、リンクコア20はこれに関与しない。
本実施形態では、このことに着目し、図23(A)にお
いてヘッダCRCが配置される第4クワドレットに、図
27に示すようにデータポインタを配置している。そし
て、ヘッダCRCが生成される期間を利用して、RDA
TAからデータポインタであるH4を取得するようにし
ている。このようにすることで、処理時間に無駄が生じ
るのを防止できるようになる。

【0196】データポインタが取得されると、ポインタ
更新回路284が、取得されたデータポインタであるH
4のインクリメントを行う。そして、アドレス発生回路
288は、図26のB12に示すように、インクリメン
トされるデータポインタに応じたRADRを発生する。
このようにして、RDATAのデータ部分がRAM80
から順次読み出される。

【0197】1つのパケットの送信処理が終了し、B4
に示すようにTCMPがアクティブになると、B13に
示すようにACK書き込み要求発生回路294がWRE
Qをアクティブにする。そして、リンクコア20からA
CK書き込みデータ&アドレス発生回路296に対して
TACKを用いて送られたACKのコードが、B14に
示すように、WDATAとして出力される。また、この
際に、ACKのコードの書き込みアドレスであるHP+
7が、WADRとして出力される。ここで、WADRが
HP+7になるのは、図27に示すようにACKのコー
ドはヘッダの第7クワドレットに書き戻されるからであ
る。

【0198】以上のようにして、ヘッダ領域のヘッダと
データ領域のデータを結合して、送信パケットを組み立
てることが可能になる。

【0199】特に本実施形態では、ヘッダとデータの結
合はDMAC40が行い、リンク20はこれに関与する
必要がない。従って、リンクコア20の回路構成や処理
を簡素化できるようになる。

【0200】また、本実施形態では、データポインタ取

得回路285が、RDATAからデータポインタ(H
4)を取得し、この取得されたデータポインタに基づき
RADRが発生し、データが読み出される。このように
することで、ヘッダと、そのヘッダに対応するデータと
を、確実に結合できるようになる。また、ヘッダとデー
タの結合処理に必要な回路構成を簡素化できるようにな
る。

【0201】4. 電子機器

次に、本実施形態のデータ転送制御装置を含む電子機器
の例について説明する。

【0202】例えば図28(A)に電子機器の1つであ
るプリンタの内部ブロック図を示し、図29(A)にそ
の外観図を示す。CPU(マイクロコンピュータ)51
0はシステム全体の制御などを行う。操作部511はプ
リンタをユーザが操作するためのものである。ROM5
16には、制御プログラム、フォントなどが格納され、
RAM518はCPU510のワーク領域として機能す
る。表示パネル519はプリンタの動作状態をユーザに
知らせるためのものである。

【0203】PHYチップ502、データ転送制御装置
500を介して、パーソナルコンピュータなどの他のノ
ードから送られてきた印字データは、バス504を介し
て印字処理部512に直接送られる。そして、印字デー
タは、印字処理部512にて所与の処理が施され、プリ
ントヘッダなどからなる印字部(データを出力するた
めの装置)514により紙に印字されて出力される。

【0204】図28(B)に電子機器の1つであるスキ
ャナの内部ブロック図を示し、図29(B)にその外観
図を示す。CPU520はシステム全体の制御などを行
う。操作部521はスキャナをユーザが操作するための
ものである。ROM526には制御プログラムなどが格
納され、RAM528はCPU520のワーク領域とし
て機能する。

【0205】光源、光電変換器などからなる画像読み取
り部(データを取り込むための装置)522により原稿
の画像が読み取られ、読み取られた画像のデータは画像
処理部524により処理される。そして、処理後の画像
データがバス505を介してデータ転送制御装置500
に直接送られる。データ転送制御装置500は、この画
像データにヘッダなどを付加することでパケットを生成
し、PHYチップ502を介してパーソナルコンピュー
タなどの他のノードに送信する。

【0206】図28(C)に電子機器の1つであるCD
-Rドライブの内部ブロック図を示し、図29(C)に
その外観図を示す。CPU530はシステム全体の制御
などを行う。操作部531はCD-Rをユーザが操作す
るためのものである。ROM536には制御プログラム
などが格納され、RAM538はCPU530のワーク
領域として機能する。

【0207】レーザ、モータ、光学系などからなる読み

取り&書き込み部（データを取り込むための装置又はデータを記憶するための装置）533によりCD-R532から読み取られたデータは、信号処理部534に入力され、エラー訂正処理などの所与の信号処理が施される。そして、信号処理が施されたデータが、バス506を介してデータ転送制御装置500に直接送られる。データ転送制御装置500は、このデータにヘッダなどを付加することでパケットを生成し、PHYチップ502を介してパーソナルコンピュータなどの他のノードに送信する。

【0208】一方、PHYチップ502、データ転送制御装置500を介して、他のノードから送られてきたデータは、バス506を介して信号処理部534に直接送られる。そして、信号処理部534によりこのデータに所与の信号処理が施され、読み取り&書き込み部533によりCD-R532に記憶される。

【0209】なお、図28(A)、(B)、(C)において、CPU510、520、530の他に、データ転送制御装置500でのデータ転送制御のためのCPUを別に設けるようにしてもよい。

【0210】本実施形態のデータ転送制御装置を電子機器に用いることで、高速なデータ転送が可能になる。従って、ユーザがパーソナルコンピュータなどによりプリントアウトの指示を行った場合に、少ないタイムラグで印字が完了するようになる。また、スキャナへの画像取り込みの指示の後に、少ないタイムラグで読み取り画像をユーザは見るができるようになる。また、CD-Rからのデータの読み取りや、CD-Rへのデータの書き込みを高速に行うことができるようになる。更に、例えば1つのホストシステムに複数の電子機器を接続して利用したり、複数のホストシステムに複数の電子機器を接続して利用したりすることも容易になる。

【0211】また本実施形態のデータ転送制御装置を電子機器に用いることで、CPU上で動作するファームウェアの処理負荷が軽減され、安価なCPUや低速のバスを用いることが可能になる。更に、データ転送制御装置の低コスト化、小規模化を図れるため、電子機器の低コスト化、小規模化も図れるようになる。

【0212】なお本実施形態のデータ転送制御装置を適用できる電子機器としては、上記以外にも例えば、種々の光ディスクドライブ(CDROM、DVD)、光磁気ディスクドライブ(MO)、ハードディスクドライブ、TV、VTR、ビデオカメラ、オーディオ機器、電話機、プロジェクタ、パーソナルコンピュータ、電子手帳、ワードプロセッサなど種々のものを考えることができる。

【0213】なお、本発明は本実施形態に限定されず、本発明の要旨の範囲内で種々の変形実施が可能である。

【0214】例えば、本発明のデータ転送制御装置の構成は、図6に示す構成が特に望ましいが、これに限定さ

れるものではない。

【0215】また本発明では、記憶手段を複数の領域に分離することが特に望ましいが、これを分離させない構成とすることも可能である。

【0216】また、本発明は、IEEE1394規格でのデータ転送に適用されることが特に望ましいが、これに限定されるものではない。例えばIEEE1394と同様の思想に基づく規格やIEEE1394を発展させた規格におけるデータ転送にも本発明は適用できる。

10 【0217】

【図面の簡単な説明】

【図1】図1(A)、(B)、(C)は、非同期転送とアイソクロナス転送について説明するための図である。

【図2】図2(A)、(B)は、ツリー識別について説明するための図である。

【図3】自己識別について説明するための図である。

【図4】図4(A)、(B)、(C)、(D)は、セルフIDパケットなどの物理層のパケットのフォーマットを示す図である。

20 【図5】IEEE1394のプロトコル構成について示す図である。

【図6】本実施形態のデータ転送制御装置の構成例を示す図である。

【図7】ヘッダ(制御情報)領域とデータ領域の分離について説明するための図である。

【図8】本実施形態の比較例の構成例について示す図である。

【図9】図8の構成によるデータ転送の手法について説明するための図である。

30 【図10】データ転送の手法の他の例について説明するための図である。

【図11】本実施形態のデータ転送の手法について説明するための図である。

【図12】図12(A)、(B)は、リンクコアとCPUとの間にランダムアクセス可能なRAMを介在させる手法について説明するための図である。

【図13】受信パケットをヘッダとデータに分離してRAMのヘッダ領域とデータ領域に格納する手法について説明するための図である。

40 【図14】ヘッダ領域に格納されるヘッダとデータ領域に格納されるデータを結合し、送信パケットを組み立てる手法について説明するための図である。

【図15】ヘッダ領域に格納するヘッダにデータポインタを含ませる手法について説明するための図である。

【図16】TAGについて説明するための図である。

【図17】図17(A)、(B)、(C)、(D)、(E)、(F)は、RAMを各種の領域に分離する手法について説明するための図である。

50 【図18】RAMの各領域のサイズを可変に制御する手法について説明するための図である。

【図19】図19(A)、(B)、(C)は、リングバッファ構造について説明するための図である。

【図20】図20(A)、(B)は、1つのパケットのヘッダやデータが格納される領域のサイズを固定にする手法について説明するための図である。

【図21】受信側の構成の一例を示す図である。

【図22】受信側の動作について説明するためのタイミング波形図である。

【図23】図23(A)は、IEEE1394規格の非同期パケットのフォーマットであり、図23(B)は、RAMのヘッダ領域に格納される非同期受信パケットのヘッダ部分のフォーマットである。

【図24】図24(A)、(B)、(C)、(D)は、ヘッダポインタとデータポインタの更新について説明するための図である。

【図25】送信側の構成の一例を示す図である。

【図26】送信側の動作について説明するためのタイミング波形図である。

【図27】RAMのヘッダ領域に格納される非同期送信パケットのヘッダ部分のフォーマットである。

【図28】図28(A)、(B)、(C)は、種々の電子機器の内部ブロック図の例である。

【図29】図29(A)、(B)、(C)は、種々の電子機器の外観図の例である。

【符号の説明】

10 PHYインターフェース
20 リンクコア
22 レジスタ
30 FIFO(ATF)
32 FIFO(ITF)
34 FIFO(RF)
40 DMAC(ATF用)
42 DMAC(ITF用)
44 DMAC(RF用)
50 ポートインターフェース
52 FIFO(PF)
54 DMAC(PF用)
56 レジスタ
60 CPUインターフェース
62 アドレスデコーダ
63 データ同期化回路

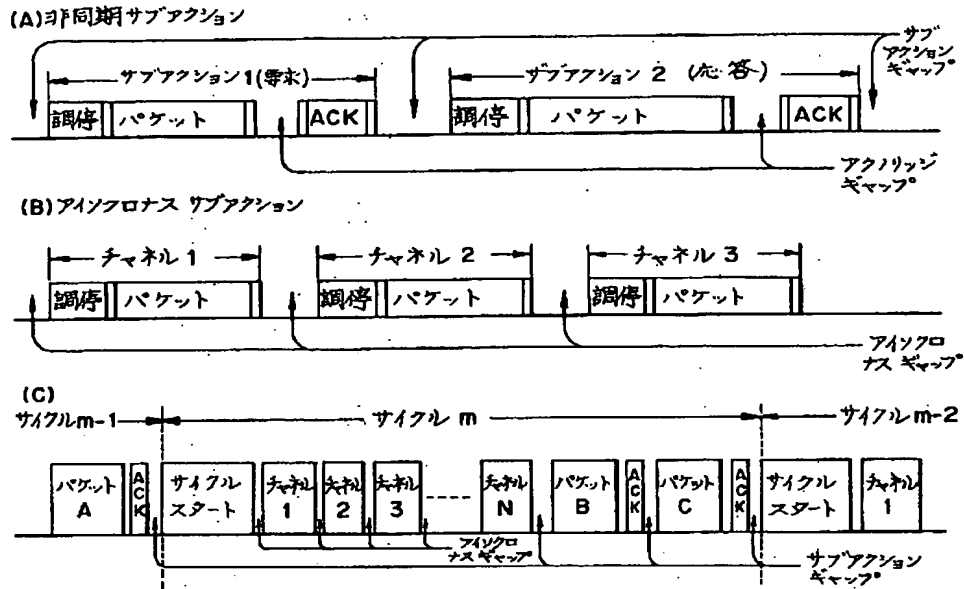
* 64 割り込みコントローラ
66 CPU
68 クロック制御回路
70 バッファマネージャ
72 レジスタ
74 調停回路
76 シーケンサ
80 RAM(パケット記憶手段)
90、92、94 バス(第1のバス)
96、98 バス(第2のバス)
100、102、104、105、
106、107、108、109 バス(第3のバス)
110 バス(第4のバス)
120 データ転送制御装置
122 PHYチップ
124 アプリケーション層のデバイス
130 バス監視回路
132 直列・並列変換回路
142 パケット診断回路
160 パケット整形回路
162 TAG生成回路
164 ヘッダ&トレイラー生成回路
166 エラーチェック回路
167 シーケンサ
168 バッファ
170 セレクタ
180 パケット分離回路
182 TAG判別回路
184 ポインタ更新回路
188 アドレス発生回路
190 アクセス要求実行回路
192 アドレス要求発生回路
280 パケット結合回路
284 ポインタ更新回路
285 データポインタ取得回路
286 tcode判別回路
290 アクセス要求実行回路
292 アドレス要求発生回路
294 ACK書き込み要求発生回路
296 ACK書き込みデータ&アドレス発生回路

*

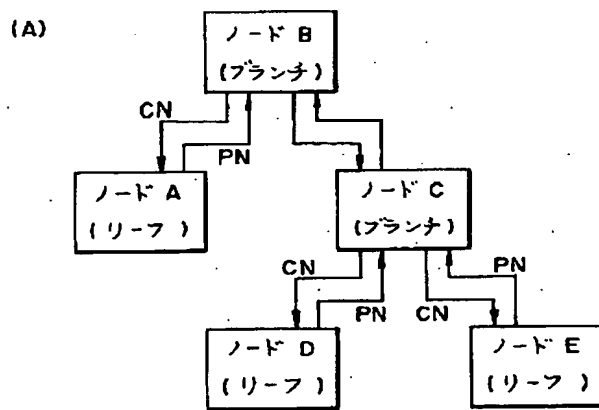
【図16】

TAG(DTAG)	意味
00	ヘッダ
01	トレイラー
10	データ
11	スタート

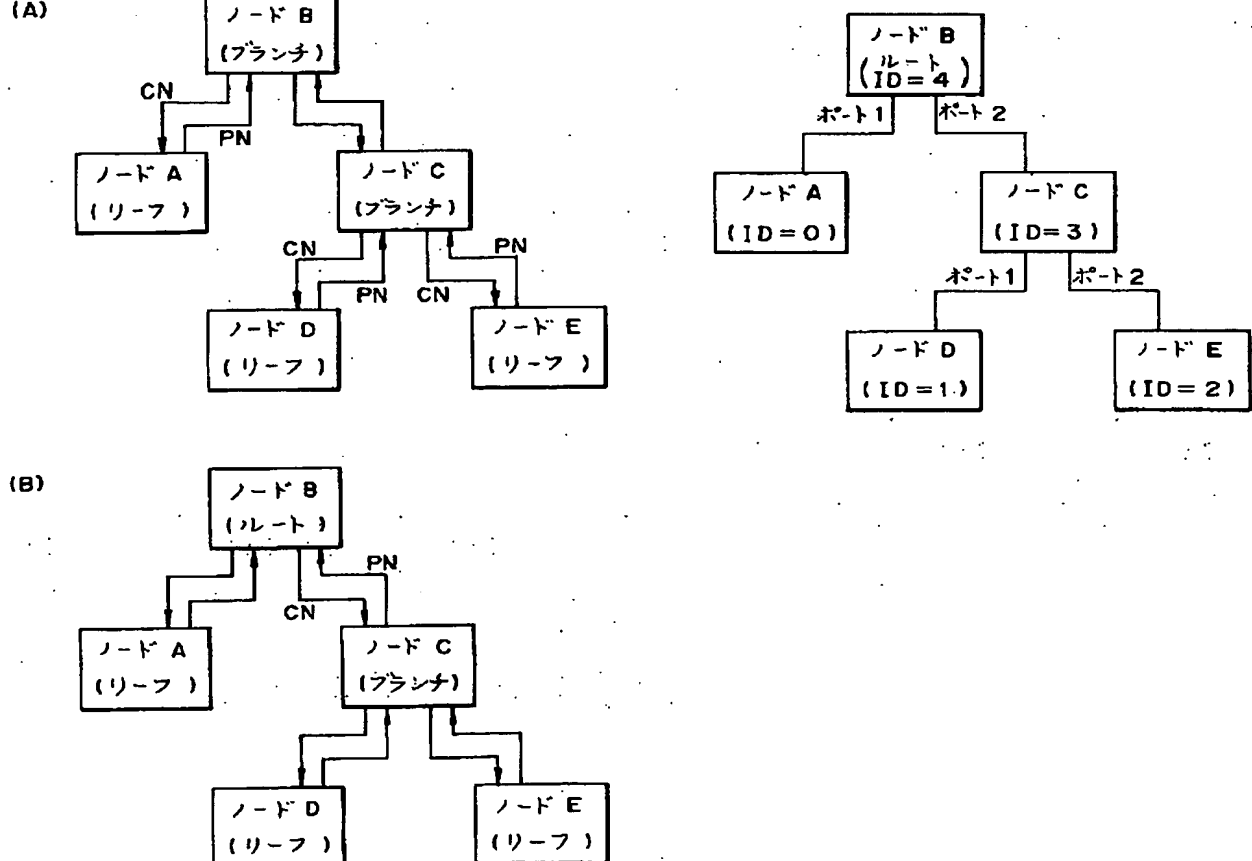
【図1】



【図2】



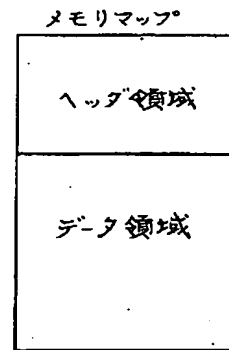
【図3】



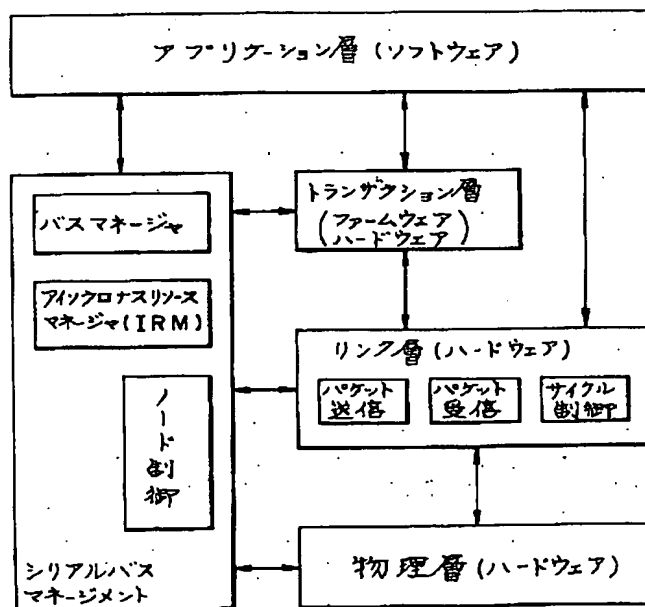
【図4】



【図7】



【図5】



【図13】

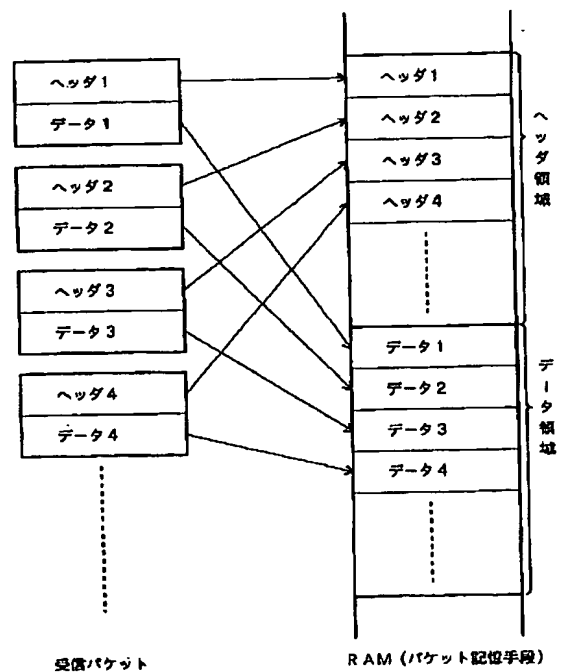
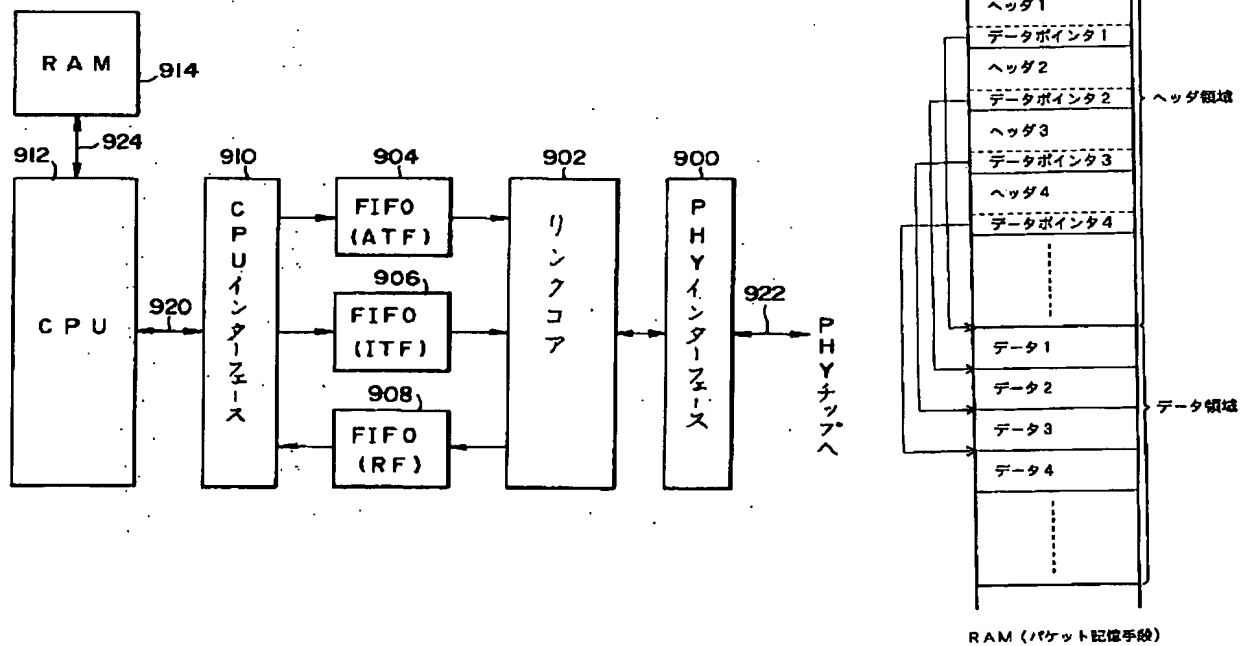
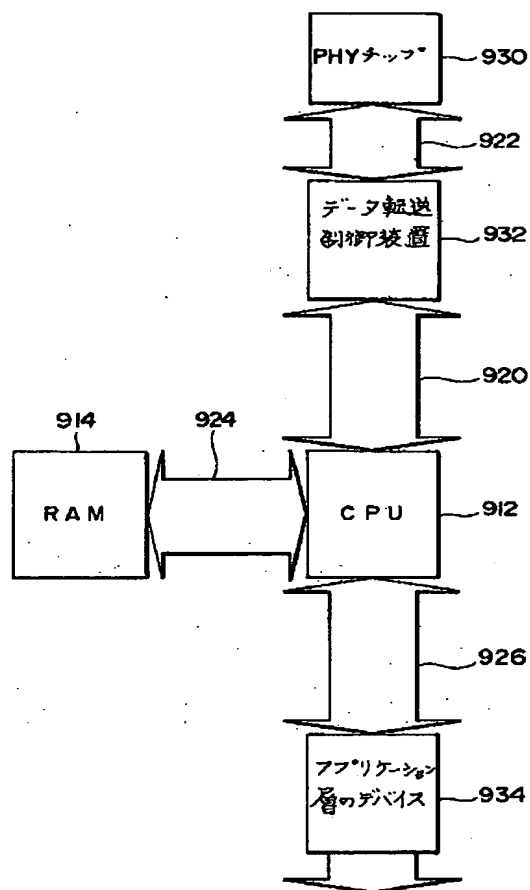


Figure 1 is a block diagram of a system architecture. The diagram shows a central CPU (66) connected to various components. At the top is RAM (60) with address lines 70, 72, 74, and 76. Below RAM are a Register (レジスタ, 72), a Backup Manager (バックアップマネージャ, 70), a Control Circuit (制御回路, 74), and a Sensor (センサ, 76). The CPU is connected to an Address Decoder (アドレスデコーダ, 62), a Data Transfer Circuit (データ転送回路, 63), a CPU Interface (CPUインターフェース, 64), and a Control Counter (制御カウンタ, 66). On the left, a Port Interface (ポートインターフェース, 50) connects to a Register (レジスタ, 56) and a FIFO (FIFO (PF), 52). The FIFO (52) connects to a DMAC (DMAC (PF用), 54). A DMAC (DMAC (ITF用), 40) connects to a FIFO (FIFO (ITF), 30). A DMAC (DMAC (ITF用), 42) connects to a FIFO (FIFO (ITF), 32). A DMAC (DMAC (RF用), 44) connects to a FIFO (FIFO (RF), 34). A Register (レジスタ, 46) is connected to the CPU interface. On the right, a Link Core (リンクコア, 20) connects to a PHY Interface (PHYインターフェース, 10) and a PHY Chip (PHYチップ, 100). A Clock Divider Circuit (クロック分割回路, 68) provides SCLK (SCLK) and HCLK (HCLK) signals. The diagram also shows various data buses and control lines connecting these components.

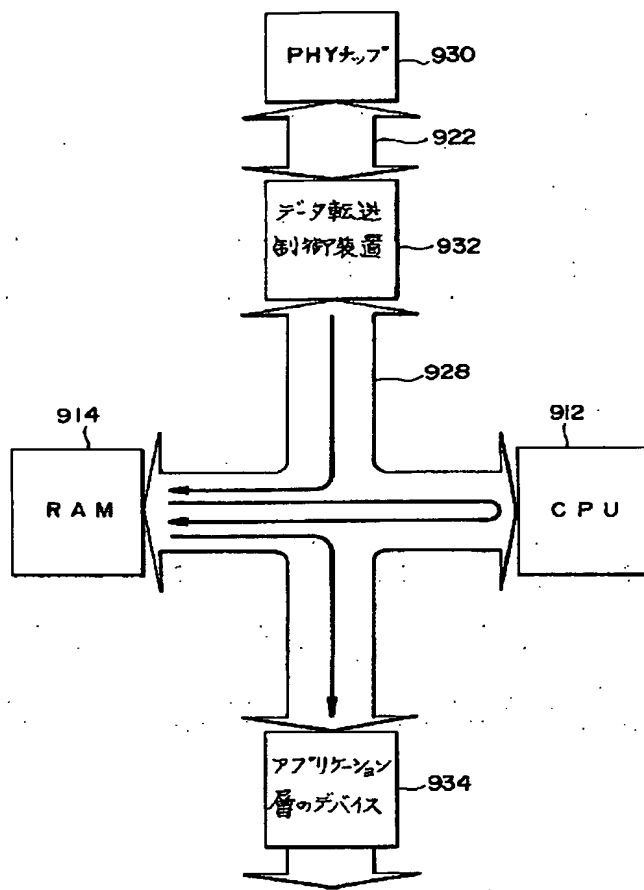
【図 15】



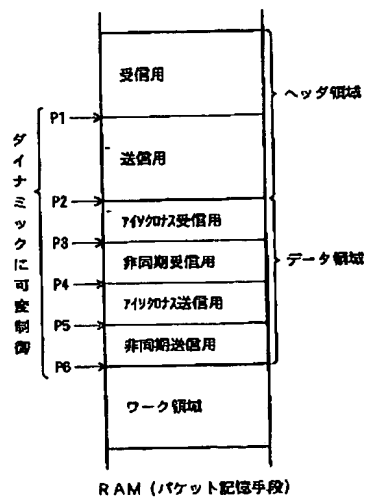
【図9】



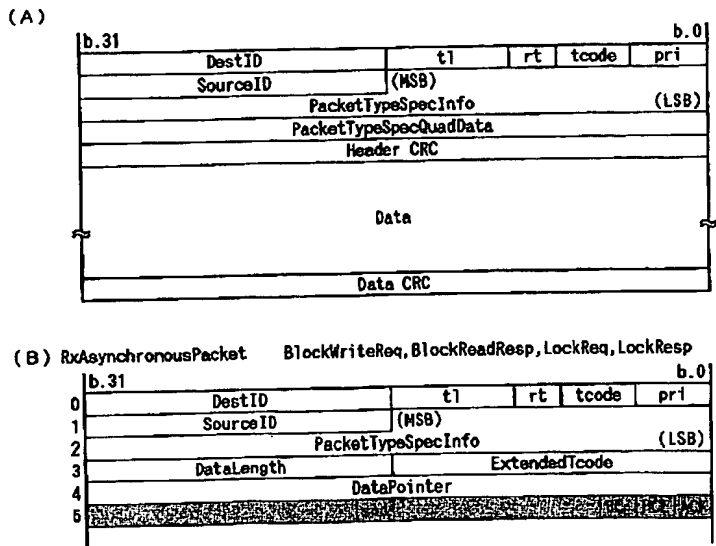
【図10】



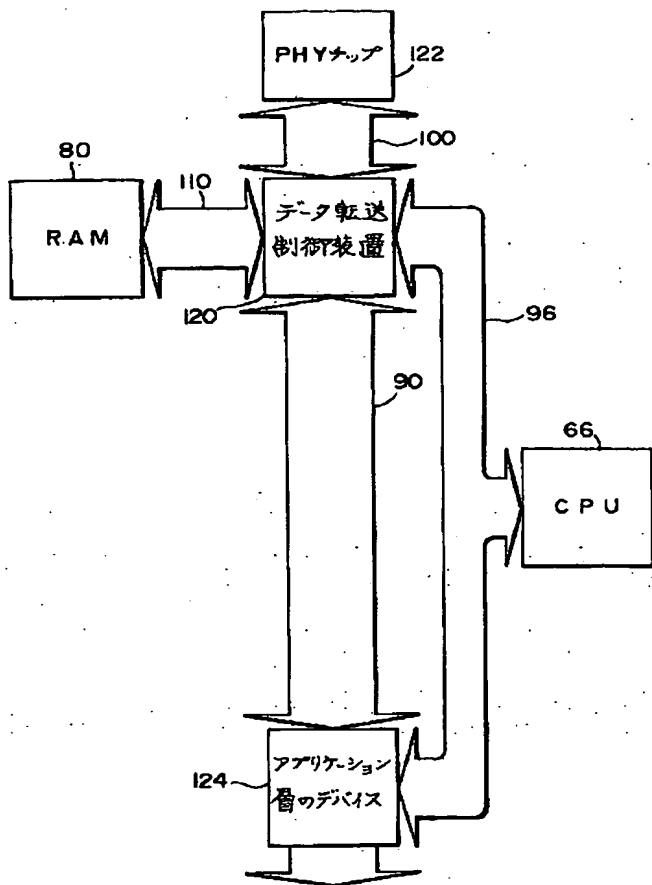
【図18】



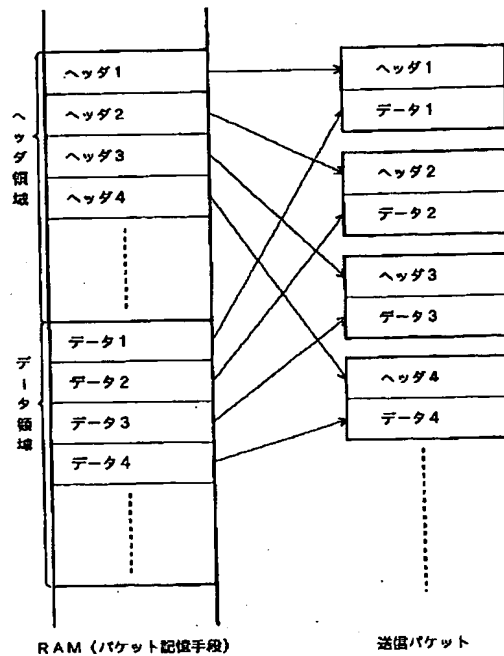
【図23】



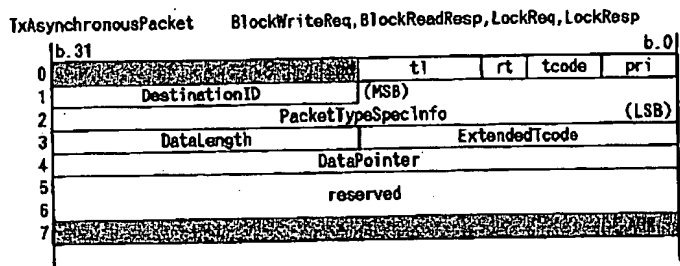
【図11】



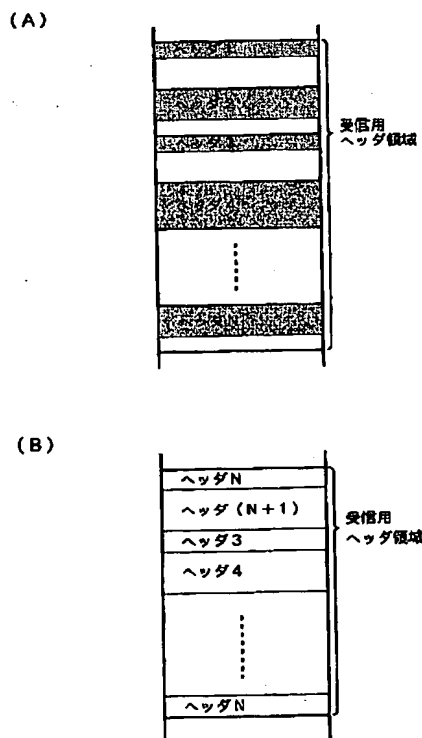
【図14】



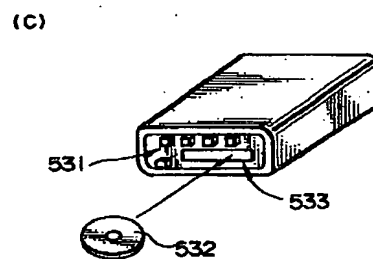
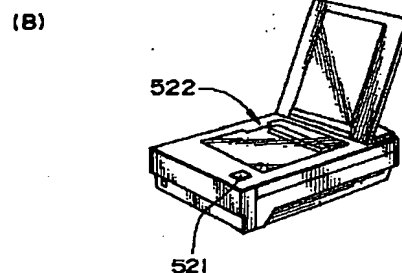
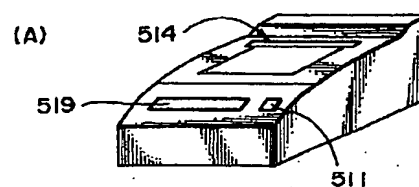
【図27】



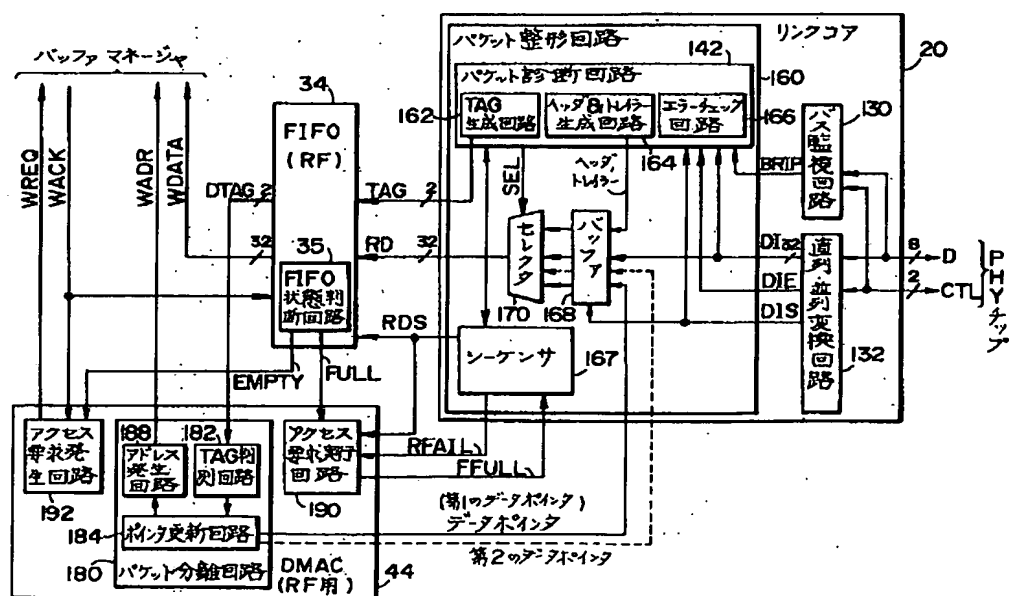
【図20】



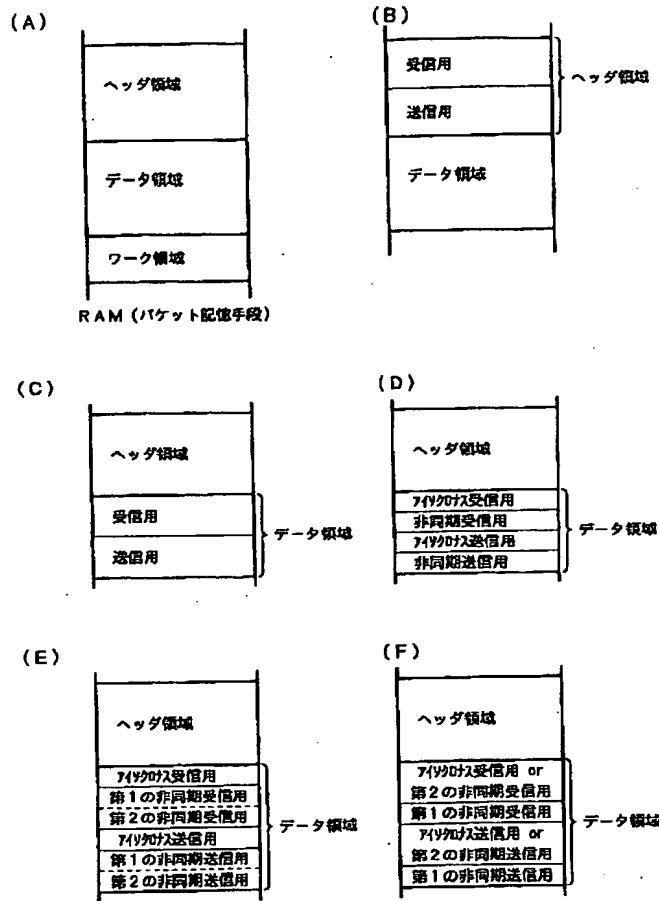
【図 29】



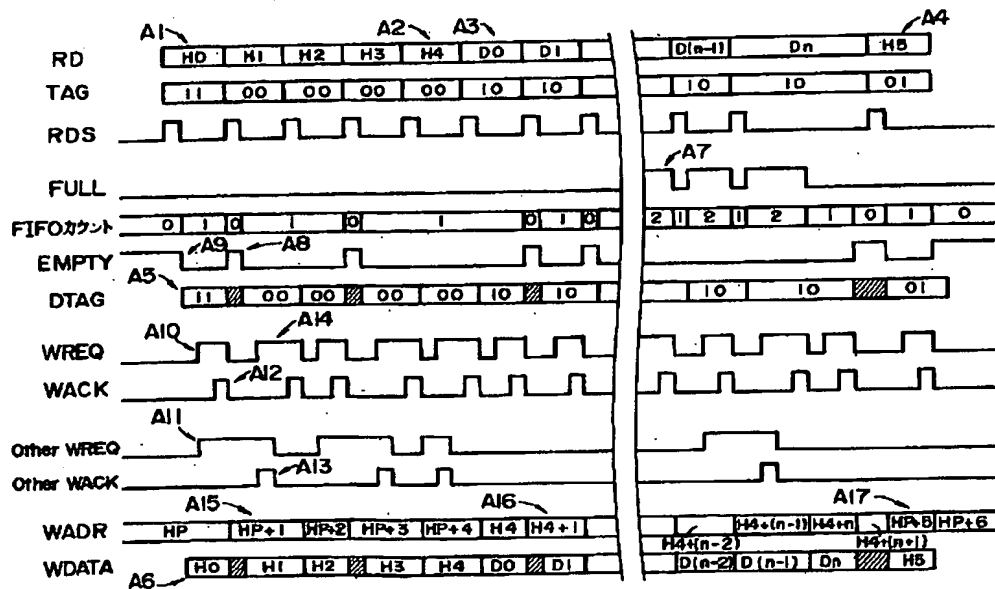
【図 2 1】



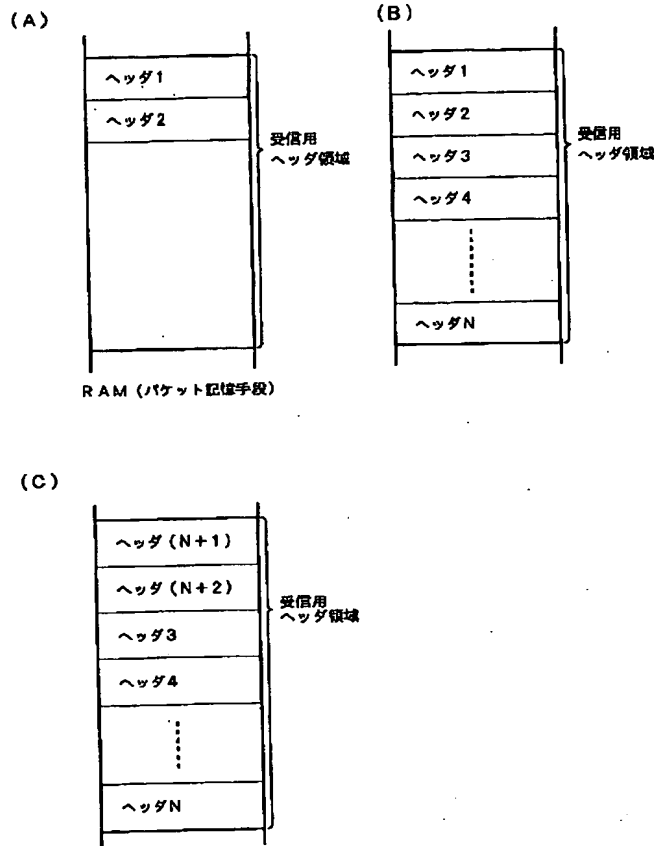
【図17】



【図22】



【図19】



【図25】

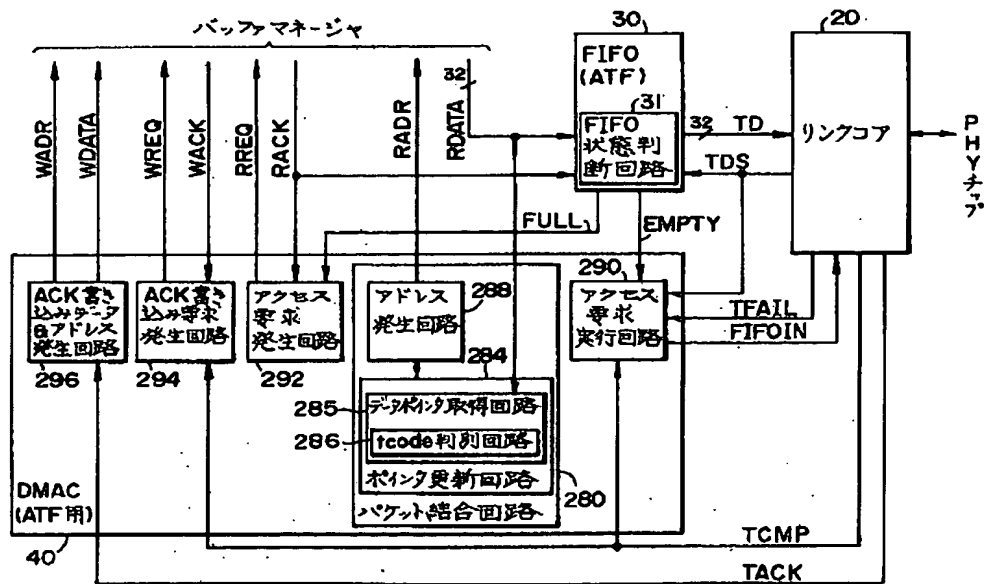


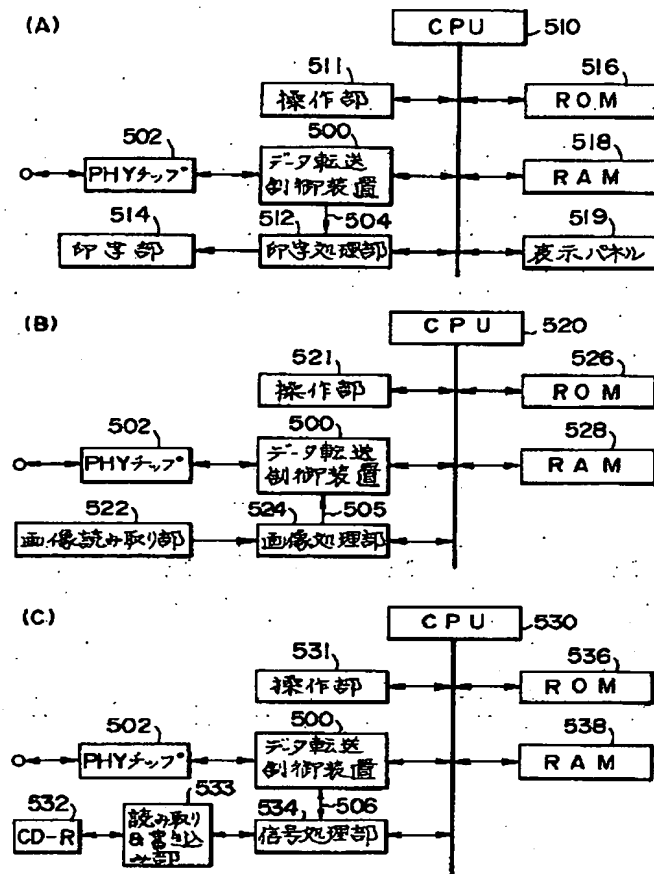
Figure 1 consists of four diagrams (A, B, C, D) illustrating memory layout for a 16-bit data bus. Each diagram shows a vertical stack of memory cells. The top section is labeled 'ヘッダ領域' (Header Area) and contains fields H0, H1, H2, H3, H4, and H5. The bottom section is labeled 'データ領域' (Data Area) and contains fields D0, D1, D2, and Dn. Diagram (A) shows a standard layout with a pointer 'HP' pointing to the start of the header area and 'インクリメント' (Increment) indicating the next address. Diagram (B) shows a layout with a 4-byte offset (HP+4) between the header and data fields. Diagram (C) shows a layout with a 4-byte offset (HP+4) between the header and data fields, but with a different data field offset (H4*(n+1)). Diagram (D) shows a layout with a 6-byte offset (HP+6) between the header and data fields.

The timing diagram illustrates the sequence of events for a data transfer. The signals shown are:

- TSTART**: Timing start signal.
- FIFOIN**: FIFO input signal.
- TD**: Data bus signal, showing data words $H0, H1, H2, H3, D0, D1, D2$ and $2n-1, 2n$.
- TDS**: Data strobe signal, with points $B1, B3, B2$ marked.
- TCMP**: Timing completion signal.
- TACK**: Acknowledgment signal, showing **ACKCODE**.
- EMPTY**: Empty signal, with point $B6$ marked.
- FIFOカント**: FIFO count signal, showing values 0, 1, 2, 1, 0, 1, 2, 1.
- FULL**: Full signal, with point $B7$ marked.
- RREQ**: Read request signal, with points $B8, B9$ marked.
- RACK**: Read acknowledgment signal, with point $B12$ marked.
- RADR**: Read address signal, showing $HP, HP+1, HP+2, HP+3, HP+4, H4, H4+1, H4+2, H4+3$ and $H4+n, H4+n+1, H4+n+2, H4+(n+3)$.
- RDATA**: Read data signal, showing $H0, H1, H2, H3, H4, D0, D1, D2$ and $Dn, Dn+1, Dn+2$.
- WREQ**: Write request signal.
- WACK**: Write acknowledgment signal.
- WADR**: Write address signal, showing $HP+7$.
- WDATA**: Write data signal, showing **ACKCODE**.

Other labels include $B4, B5, B11, B13, B14$ and $CRCエラー$ (CRC error).

【図28】



フロントページの続き

(72)発明者 神原 義幸
長野県諏訪市大和3丁目3番5号 セイコ
ーエプソン株式会社内

Fターム(参考) 5K032 AA02 AA03 CD01 DB19 DB24